Institute of Cybernetics, Tallinn University of Technology
Institute of the Estonian Language

The Second Baltic Conference on HUMAN LANGUAGE TECHNOLOGIES

Proceedings

April 4 - 5, 2005, Tallinn, Estonia

Editors:
Margit Langemets, Priit Penjam

Tallinn, 2005

Supported by eVikings II, Intenational Speech Communication Association,
Tilde, Estonian Ministry of Education and Research.

Cover design by Piia Ruber.

Printed by Tallinna Raamatutrükikoda.

ISBN 9985-894-83-9

Content

Programme Committee

TOM BRØNDSTED,
Aalborg University, Denmark

BJÖRN GRANSTRÖM,
Royal Institute of Technology, Sweden

MARE KOIT,
University of Tartu, Estonia

MATTI KARJALAINEN,
Helsinki University of Technology, Finland

STEVEN KRAUWER,
European Network of Language Excellence (Elsnet), The Netherlands

RŪTA MARČINKEVICIENĖ,
Vytautas Magnus University, Lithuania

EINAR MEISTER,
Institute of Cybernetics, Estonia

INGUNA SKADIŅA,
University of Latvia, Tilde, Latvia

ANDREJS SPEKTORS,
University of Latvia, Latvia

LAIMUTIS TELKSNYS,
Institute of Mathematics and Informatics, Lithuania

ANDREJS VASIĻJEVS,
Tilde, Latvia

ÜLLE VIKS,
Institute of the Estonian Language, Estonia

STEFAN WERNER,
University of Joensuu, Finland

Welcome Message

On behalf of the conference organisers, the Institute of Cybernetics at the Tallinn University of Technology and the Institute of the Estonian Language, I would like to welcome all participants to the Second Baltic Conference on Human Language Technologies – HLT'2005, which is being held in Tallinn during April 4 - 5, 2005.

This conference is the follow-up of the First Baltic Conference on HLT, held in Riga, Latvia, almost a year ago. Within this short period of time the Baltic States have gone through a significant change – they have become full members of the European Union. This event represents the start of a new era that brings with it a large number of potential opportunities to participate in all areas and activities of a new multi-cultural and multi-lingual community. How to survive in a multilingual Europe without losing one's language and identity is certainly a crucial issue for all Baltic countries. We – the participants of this conference – understand that the development of efficient technologies for the processing of human languages can help languages survive so as to prevent Gutenberg's effect from taking place in the computer age. Therefore, conferences like this are of a great importance since they provide a forum for the sharing of new ideas and recent advances in human language processing. They also promote interdisciplinary co-operation between the research communities of computer science and linguistics from the Baltic countries and the rest of the world.

The scientific programme of the conference consists of four plenary sessions including seven presentations by world renown scientists, as well as 20 oral and 28 poster presentations selected by the international programme committee chosen from 60 submitted abstracts.

The Tutorial Day on April 6 offers two parallel sessions, one in spoken language and the other in written language technology. Each session includes three lectures by established scholars, and ends with a joint session that integrates the two fields.

I would like to thank the supporters of the conference:

- eVikings II – a FP5 project IST 2001-37592
- International Speech Communication Association (ISCA)
- Estonian Ministry of Education and Research
- Tilde

I wish you all a successful conference and a very pleasant stay in Tallinn.

Einar Meister, Ph.D.
Chair of HLT'2005

I Plenary Papers

EXPRESSIVE SPEECH SYNTHESIS: WHAT IS THE GOAL?

Nick Campbell
ATR Network Informatics Lab (Kyoto, Japan)

Abstract

This paper describes a recent attempt to define expressiveness in the context of speech synthesis. With concatenative techniques, we have overcome the barrier of portraying extra-linguistic information in speech synthesis, by using the actual voice of a recognisable person as a source for units, but we still face the barrier of expressing the variety in the types of speech that a person might use in everyday social interactions. Paralinguistic modification of an utterance portrays the speaker's affective states and shows his or her relationships with the speaker through variations in the manner of speaking, by means of prosody and voice quality. These inflections are carried on the propositional content of an utterance, and can perhaps be modelled by rule, but they are also expresssed through non-verbal utterances, the complexity of which may be beyond the capabilities of current synthesis methods.

Keywords: Communication, Expression, Affect, Emotion, Social Interaction, Non-Verbal, Speech Synthesis, Human-Interface Technology

1. Introduction

The computer synthesis of speech has been a goal of computer scientists and speech technologists for more than half a century (1; 2), yet neither linguists nor phoneticians have yet achieved a comprehensive definition of the full range of variations and uses of speech as a means of human communication and social interaction.

Much of the research into human language has been based on the analysis of written texts. When spoken language has been considered, it has been treated either as a system of sounds or as a media-transformed version of text; to be analysed in a written form through the use of transcriptions. This is understandable, since the technology for analysing oral interactions has until recently been both expensive and lacking in portability. 'Speech' is not well understood from the standpoint of 'communication'.

Similarly, conversation analysis has a long history of research, but again, in the majority of cases, it is the texts of the conversations that has formed the basic material for study. The actual sounds of the speech have been considered as of secondary importance to the content. *What you say* has been treated as more important than *how you say it*; but whereas this may well be the case for formal announcements, it is rarely so in casual interpersonal interactions, where phatic communion is as important as propositional content, if not even more so.

More recently, we find many comprehensive resources of spoken material available to researchers, thanks largely to the efforts of the speech recognition community to provide training material for their statistical engines. In the early days of speech recognition research the emphasis was more phonetic — categorising the basic speech sounds by use of Hidden Markov Models, and using triphone-contexts to define elemental phones to be interpreted in conjunction with the use of a language model in order to convert sound sequences into words for recognition.

Prosodic variation in speech was largely ignored, because the technology provided word candidates regardless of the speaker-specific or utterance-specific variations. The texts could be understood without resource to prosodic knopwledge, whch was thought to function primarily as a support for syntactic and semantic information encoded in the text. Directed-speech, rather than casual conversation was the norm in such research.

The emphasis in speech data collection was on maximising speaker numbers in order to produce speaker-independent models, rather than on modelling the variations in the speech of a particular individual across time. Developments in recognition technology were in the direction of whole-word modelling and in improvements to the statistical language models, but the assumption of a strong dependence between component phones and consequent word sequences remained. Recognition performance was and still is evaluated in terms of the number of words correctly transcribed. The assumption that words represent speech has been largely unchallenged, and the possibility that the same utterance can carry different meanings according to its pronunciations largely ignored.

Similarly for speech synthesis research, based on the early assumptions of synthesisers functioning as reading machines, the primary focus has been on the conversion of text sequences into sound sequences. From word-based input to speech output, the flow of processing is concentrated on predicting the sounds required to represent the word sequence in order to present the same propositional content in a different medium. A given word is given different pronunciations depending on its context in an utterance or on the syntactic structure of that utterance, but very little attention has yet been paid to the function of non-verbal utterances in speech.

Analysis of a very large corpus of natural conversational speech has shown that more than half of the utterances have minimal propositional content and that they function instead to establish speaker-listener relationships and to express the speaker's affective states for phatic communication in way that cannot be transcribed into written text. This paper tackles the issue of how to synthesise such non-verbal, phatic utterances.

2. Corpus-based Speech Synthesis

Looking back across the long history of speech synthesis research, we can see in retrospect a clear evolution from the modelling of phonetic states to the modelling of utterance characteristics. The pioneering work of Gunnar Fant in Sweden (3; 4) and Dennis Klatt and his colleagues at MIT (5; 6) in the US was founded on a phonetic view of speech as a sequence of phones, modulated by prosody to represent syntactic and semantic content. Joe Olive (7; 8; 9), Osamu Fujimura, and their colleagues at Bell Labs made a significant contribution by showing that the dynamics of the transitions between the phones carried much more information than an interpolated sequence of steady-state representations of phone centres. Yoshinori Sagisaka in Japan (10) extended this paradigm shift by concatenating non-uniform sequences of actual speech taken from readings of the most common

5000 words of the language. It became clear that the information carried in the dynamics of the speech far outweighed that of the supposed phonetic centres.

Although text can be well represented by a sequence of invariant letters, speech sounds are not invariant. They depend heavily on the various contexts of their phonation (11) . My own work extended the above trend by incorporating prosodic contexts among the selection criteria for units for concatenation from a speech corpus (12; 13; 14). Although a small step in terms of processing, this allowed us to remove the signal processing component from the synthesiser and to use the speech segments intact, without resorting to potentially damaging prosodic modification. By simply concatenating phone-sized segments which had been selected according to both phonetic and prosodic contextual criteria, we were able to faithfully reproduce the voice and given speaking-style of a speaker and speech corpus (15; 16).

The early generations of speech synthesisers were soon able to reproduce the linguistic content of a message. The developments described above resulted in an ability to reproduce extra-linguistic content; i.e., the speaker-specific characteristics. However, the remaining paralinguistic aspects of speech are still poorly modelled. Speech synthesis can function effectively when presenting information by use of a given voice, but it cannot yet perform in a conversational context where the expression of affect and the management of discourse flow take on a greater importance.

3. Emotion and Speech

The latest trends in speech synthesis research can perhaps be summarised by one word: ‘emotion’ (17; 18; 19) . The poor take-up of speech technology in general by members of the public is currently attributed, by both the synthesis and recognition communities, to a lack in the ability to process emotion in the speech.

While it may well be true that current speech technology is lacking a ‘human’ component, is this really best described by the term ‘emotion’? I disagree. Or rather, I believe that what many people understand by the wider colloquial application of this term is not well represented by the more limited technical application of the term, as characterised by the ‘big-six’ emotions of psychological research as illustrated by Ekman and his colleagues (20).

Most speech technology research is based upon the analysis and modelling of speech corpora. These corpora are generally produced in controlled conditions; whether in a recording studio, using the voices of professional speakers to produce ‘clean’ data, or over the telephone, using the voices of many speakers to produce ‘representative’ data. The demands of scientific research and technological developent require that we balance the speech data so that they will be representative of the aspects of speech which we wish to reproduce. These controls can take the form of ‘phonetic balance’, from reading of carefully produced sets of sentences so that each phone is presented in every context of possible use, or of ‘sociological balance’ so that each sector of the community is ‘equally’ represented, or of ‘content balance’ so that all speakers produce a common set of desired utterance types.

The drawback with the above ‘scientific’ constraints is that we only see what we originally intended to look at. That is, the data that we produce for research are representative of the aspects of speech that are generally considered to be important at a given stage of the evolution of the technology, but they are not necessarily representative of the ways that ordinary people use speech in the everyday contexts of social interaction. When confronting researchers with this dilemma, whether in a review of a submitted journal paper or in casual conversation, we often meet the response “Well, what else can we do?”. It appears that many of us are aware of the drawbacks of this approach but that we nonetheless continue to follow in the footsteps of our predecessors. Such is the path of scientific research.

So how does this affect the representation of emotion in speech? The chain of logic is as follows: (i) emotion is poorly represented in current speech processing, so (ii) emotionally charged speech data should be collected, (iii) the texts must be balanced so that scientific comparisons can be made, so (iv) semanticaly neutral sets of sentences should be produced under various emotions, so (v) actors are recorded producing each sentence in every emotional state, then (vi) perception tests are carried out to ‘validate’ the data, and (vii) subsequent analyses confirm the clear acoustic characteristics of the different ‘emotions’. This is a very logical progression but it results in a corpus of stereotypical expressions that may have very little to do with how ordinary people vary their speech in actual social interactions.

Actors are trained to project what will be readily perceived as a given emotion, and listeners in the perception tests are given forced-choice between alternatives which restrict them from qualifying or elaborating on their ‘peceptions’ in any way. Furthermore, the ‘emotions’ that are almost always produced for such data tend to be simple basic ones: sadness, fear, anger, and joy, rather than the more subtle and complex states than result from the interaction of emotions arising from social interaction. It is rare in everyday life for us to experience fear and joy to the extent that they are produced in such ‘balanced’ data.

Despite the popularity of the keyword ‘emotion’ in current speech technology research, the question remains of whether this is in fact the proper direction in which to further our work. Are not ‘attitudes’ more relevant to spoken interactions? Perhaps we experience boredom or frustration more often that we experience sadness and joy? And show interest more often than we show anger. These more complex expressions of affective states and social relationships are far more common than the expression (or even the experience?) of the basic emotions as illustrated by Ekman in his work on facial expression. Certainly for the use of speech synthesis or recognition in social situations, we need also to be able to reproduce and recognise the more subtle expressions of speaker states and relationships — not just those deliberately produced on demand, but also those which are revealed in spite of a veneer of civilised self-control.

4. A Conversational Corpus

In order to discover what the more likely distributions of affective or emotional expressions might be, we produced a corpus of everyday conversational speech, which has been reported in detail elsewhere (21; 22). In order to overcome Labov’s well-known Observer’s Paradox, wherein the presence of an observer or a recording device influences the productions of the observed person, we persuaded our subjects to wear small head-mounted studio-quality microphones for extended periods while going about their normal everyday social interactions over a period of about five years.

These volunteers were paid by the hour of speech that they produced for us, and a further group were paid to transcribe and annotate this speech data in fine detail. The transcriptions were produced in plain text rather than phonetic coding, but care was taken to transcibe every utterance exactly as it was spoken, with no effort made to ‘clean-up’ the transcriptions or correct the grammar.

Transcribers were encouraged to break the speech into the smallest possible utterance chunks by use of a notional yen-per-line payment policy. In spite of this, many single ‘utterances’ included several tens of syllables, being expressed as a single breath-group. The text of the transcriptions from one speaker, if printed end-to-end as a solid block of text in book form would fill 35 volumes, and if printed one-line-per-utterance, would probably exceed 100 volumes.

The majority of speech utterances in this corpus were single phrases; ‘grunts’, or phatic sounds made to reassure the listener of the speaker’s affective states and discoursal intentions (23; 24). Laughs were very frequent, as were back-channel utterances and fillers[1], but approximately half the number of utterances transcribed were unique. These typically longer utterances can perhaps be well handled by current speech synthesis techniques, since the text carries the brunt of the communication, but the shorter ‘grunts’ require a new method of treatment.

The word ‘grunt’ carries implications of pre-human or even animal behaviour, but I believe that it is the most appropriate term for the type of phatic communication that takes the place of mutual grooming in human society (25). As well as the frequent “ummm”, “ahhh”, “yeah”, “uh-uh”, etc., I include the use of such phrases as “good morning!” and “did you sleep well?”, “see the game last night?”, etc., which are used when social rather than propositional interactions are normal. They float to the top of the multigram dictionary (26) by dint of their frequent occurrence, but most can be characterised by the flexibility and variety of their prosody. None can be interpreted from the plain text alone. Perhaps these sounds are among the oldest forms of spoken language? In numerical terms, they account for more than half of the conversational corpus.

On the basis of the above distinction, we categorise the corpus utterances in terms of I-type and A-type functions; the former for the conveyance of information, the latter for the expression of affect (27; 28). A framework was proposed (see Figure 1 for an illustration) which describes the two-way giving and getting of I-type and A-type information subject to speaker-state and listener-relationships. For simplicity in a speech synthesis application, we propose four levels of each:

- Self (the speaker herself)
  - Mood: the speech is ‘brighter’ if the speaker is in a ‘good mood’ (two levels: plus, minus).
  - Interest: the speech is more ‘energised’ if the speaker is interested in the conversation (two levels: high, low).
- Other (her relationships with the interlocutor)
  - Friend: the speech is ‘softer’ if the listener is a friend (two levels: close, distant).
  - Place: the speech is more ‘intimate’ if it takes place in a relaxed environment (two levels: relaxed, formal).

[1] I use the word ‘filler’, since it is common parlance, though I strongly object to the implication that a gap exists in the interaction which is being ‘filled’. I believe that these slots in the communication process serve a very important function as places where non-linguistic (affective) communication can occur.

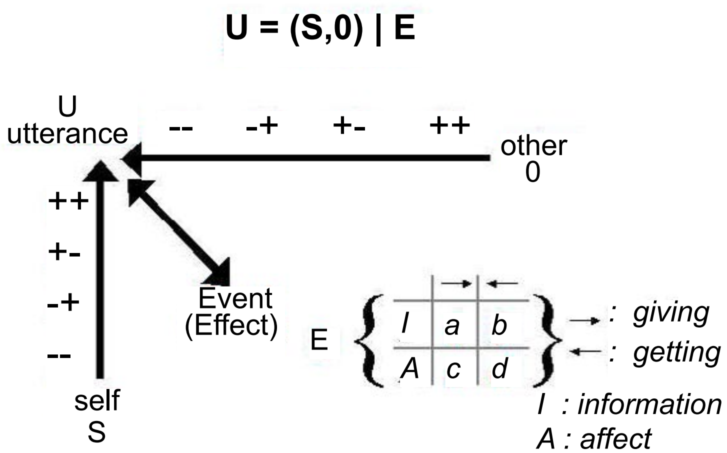


Figure 1: A framework for specifying the characteristics of an utterance according to speaker-state, relationship with the listener, and speech-act type.

Any given utterance will be realised subject to the above constraints. The challenge to synthesisers for conversational speech is to allow the user to specify such constraints simply and easily. In the case of A-type utterances, the framework is more important than the text, which can be relatively freely specified so long as it fulfills the desired social function of the utterance, as we will see below.

5. Interfaces for Expressive Speech Synthesis

As explained above, we can consider that there are two types of utterance in common use in conversational speech; one for transmitting propositional content (I-type), and the other for expressing affect (A-type). While existing speech synthesis technology is arguably quite adequate for the former, the subtlety of prosodic expression and voice-quality (laryngeal phonation settings) required for the latter is beyond the capability of most present systems.

While research is being carried out into signal processing techniques for modifying the voice-source settings, we have yet to find a method that is capable of also matching the sub- and supra-glottal conditions so that a realistic coherent sound can be produced. At present, any modification of the speech signal results in a perceptible degradation which is unacceptable, given that we are trying to control fine modifications in vocal setting, such as tenseness and laxness of the voice (29; 30). The vocal tract can perhaps be adequately modelled as a series of resonant tubes for the purpose of reproducing the basic speech sounds, but for the fine details of airflow required to reproduce the subtle nuances of expression in conversational speech, the model becomes excessively complex.

While not necessarily implying that such a large corpus would be necessary for conversational speech synthesis in different voices or languages, we were able to use the ESP corpus as a test case of what might be possible for concatenative synthesis in the future. Given 5-years of one person's daily conversational speech, we were interested to discover the extent to which the 6th year's speech might be contained within such a corpus.

Our first task was to reduce the text into fundamental units, since segmentation into phone-sized units is no longer necessary when whole utterances are included in many varied forms, each having different prosodic characteristics, as candidate units. For this we used a form of multigram analysis to determine on statistical grounds the common collocations of frequently-occurring sound sequences in the corpus. This analysis results in a dictionary of various-length sequences and a set of probabilities for each so that a subsequent Viterbi process based on the EM algorithm can determine the optimal sequence of segments for any given target utterance.

This process in fact provides a dictionary of frequently used sound sequences, or a personalised lexicon independent of any linguistic criteria, that models the common speech patterns of the corpus speaker. Favourite phrases and specific lexical sequences (e.g., adjective-noun groups) tend to be included as intact units with high probabilities in such a dictionary, while shorter patterns with even higher probabilities represent the frequent phonetic sequences (or articulatory gestures) of a given speaker. At the lowest level, single phone-sized sounds are indexed to ensure that any possible sequence of sounds can be generated.

By use of such statistically-determined non-uniform segments for concatenation, whole phrases can be linked by units representing common articulatory gestures so that a high level of naturalness, retaining the speaker-characteristics, can be maintained in the resulting synthesised speech. As we saw above though, more than half of the utterances can be expected to occur intact, as whole phrases, and can then be subcategorised according to their prosodic and voice-quality characteristics for the common A-type utterances. With so large a corpus, the task becomes one of selecting the appropriate acoustic realisation of a given phrase rather than that of creating such a phrase out of smaller component segments.

In parallel with the problem of determining optimal unit size, is the equivalent problem of how to specify such units for input to the synthesiser. Plain text is no longer appropriate when the intention of the speaker is more important than the lexical sequence of the utterance. Instead, we need to enable the user to quickly access a given corpus segment by means of higher-level intention-related constraints.

Figure 2 shows a recent prototype for such a speech synthesis interface. 'Chakai'[2] allows for free input (by typing text into the white box shown at bottom-centre) as well as the fast selection of various frequently-used phrases and, in addition, an icon-based speech-act selection facility for the most common types of 'grunt'. This format enables linking to a conventional CHATR-type synthesiser for creation of I-type utterances not found in the corpus, while providing a fast, three-click, interface for common A-type utterances which occur most frequently in ordinary conversational speech.

[2] The name, not unrelated to CHATR is composed of two Japanese syllables, meaning tea-meeting, an event during which social and undirected chat is common.

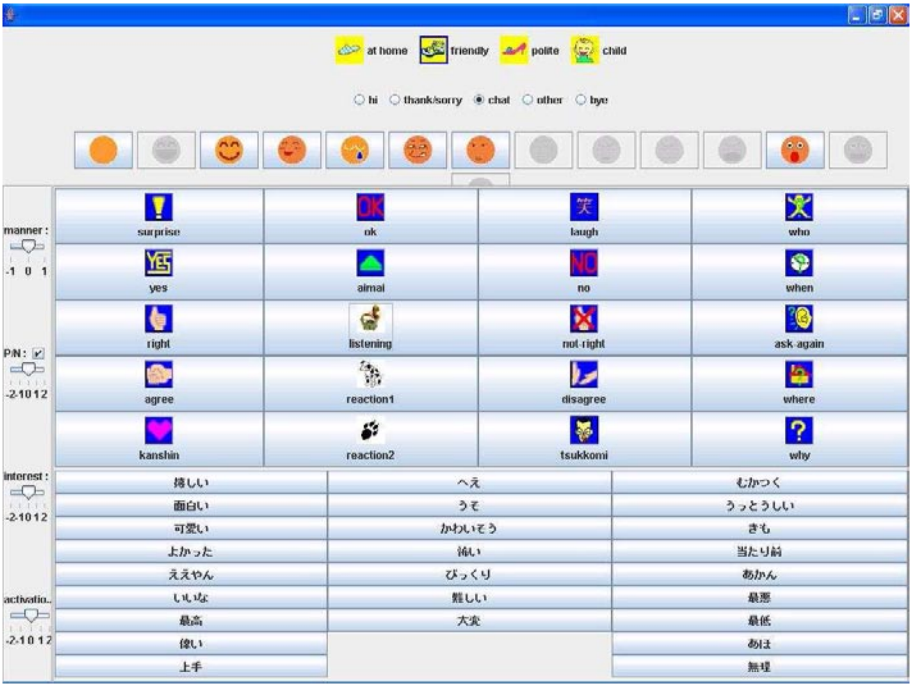


Figure 2: The Chakai Conversational Speech Synthesis interface. By clicking on a speech-act icon, a choice of emoticons is dispayed in the upper section of the screen according to corpus availability, from which an utterance having the appropriate speech characterstics can be selected. Utterances are selected at random from among those in that same category within the corpus so that subsequent selection of the same combination will provide natural variety without unnecessary repetition.

The selection of whole phrases from a large conversation-speech corpus requires specification not just of the intention of the phrase (a greeting, agreement, interest, question etc.,) but also of the speaker's affective state (as desired to be represented) and the speaker's long- and short-term relationships with the listener at that particular time.

Chakai can be used in almost real-time for conversational interaction. When initiating a topic, typed input is required, and this is presently too slow, but when showing interest or 'actively listening', then different grunts can be produced to encourage the speaker, challenge her, show surprise, interest, boredom, etc., by simply clicking on the icons. The initial frame presents the user with a choice of four listener types: friend, family, stranger, or child, with adjustable bars for setting the activation of the Self and Other constraints. The following screen allows selection of different forms of greetings, sub-categorised according to occasion (e.g., morning, evening, telephone, face-to-face, initiation, reply etc.,) with an adjustable bar for setting the intended degree of activation or 'warmth of greeting' before the penultimate button-press. When these criteria are selected, the different types of speaking style representing available utterances in the corpus are presented as a row of activated smiley-faces (top of the figure) from which the user can select the closest to their intended interactional function. No lexical-based selection or keyboard entry is offered, as the function and constraints will determine the text automatically from what is available in the corpus for that particular speaker.

The following screen (shown in the figure) is for the core part of the conversational interaction. Icons are arranged in 4 rows, with questions on the right (who, where, why, when, etc.,) and positive, neutral, and negative grunts arranged in three columns on the left of the screen. The vertical dimension here is used for degree of activation. We have tested this interface in actual conversations, and a trained operator can use it in real-tme to sustain a conversation for extended periods of time.

By splitting utterances into three types, we have greatly facilitated the selection process. I-type utterances, being largely unique since they are so content-dependent, have to be laboriously typed in. Frequent phrases which are text-specific can be selected and a choice of speaking styles is offered via the smiley-face icon layer. Grunts, which are the most common type of utterance in casual speech, are fastest to produce. An utterance can be produced simply by clicking on the type and its qualifier. The corpus has been pre-annotated for the significant dimensions of selection so the actual code that produces the segments can be very simple. And since it is often the case that whole-phrase segments are concatenated with short pauses between them, the naturalness of the resulting speech can be absolute. No processing is required, thanks to the number and variety of utterances in the corpus.

Clearly, this propotype does not represent the full final version, and it will require several generations of trial and evolution before an ideal conversation-device is realised, but we are satisfied that it well represents the problem that we are trying to solve. The user, whether handicapped or healthy, human or robot, should not have to specify the text of a conversational grunt, whether it be "yes" or "good morning" and then also have to describe its prosody or purpose. These are secondary characteristics of speech. They depend on the higher-level constraints of discourse context and speaker-intention just as the fine acoustic characteristics of CHATR segments depend on the phonetic and prosodic environment in which they occur. By knowing these dependencies and their interactions, we are able to simplify the process of selection and thereby to improve both the functionality and the quality of the synthesis process.

6. Conclusion

This paper has introduced some of our most recent work on the synthesis of conversational speech, and shown that the challenges presented by this type of task are qualitatively different from those of traditional speech synthesis for the transmission of propositional content. We have found from our analysis of a very large natural-speech corpus that at least half of the utterances in interactive conversatioal speech are not well represented by their text alone and that they depend upon specific prosodic characteristics, such as tone-of-voice, realised by differences in laryngeal phonation quality, that can not easily be reproduced by signal processing techniques. The paper has also described our initial attempts to utilise the corpus for concatenative speech synthesis, and has presented a prototype user-interface that allows input acording to speech-act intention, using constraints representing the primary contextual influences on speaking-style, so that a conversational utterance can be produced rapidly with minimal input from the user. For the phatic utterances that are a characterisatic of informal and social speech, this interface allows text-free input, since an appropriate phrase is selected from the corpus according to the higher-level constraints automatically. This work is still experimental, and the paper should not be taken to imply that the methods presented here are necessarily the best for a commercial speech synthesis system, but it presents them as an illustration of the problem rather than of its solution. We are lucky to have such a speech corpus at our disposal, but replicating it for another language or subculture would require considerable extra work.

References

[1] Holmes, J.N., Mattingley, I.G. & Shearme, J.N., ”Speech synthesis by rule”, Language and Speech 7, 127-143, 1964.

[2] Mattingly, I.G., “Experimental methods for speech synthesis by rules”, IEEE Trans. AU 16, 198-202, 1968.

[3] Fant, G. “Acoustic Analysis and Synthesis of Speech with Applications to Swedish”, Ericsson Technics 15, 3-108, 1959.

[4] Carlson, R. & Granstrom, B., “A text-to-speech system based entirely on rules”, Proc. IEEE-ICASSP76, 686-688, 1976.

[5] Allen, J., Hunnicutt, M. S. & Klatt, D.H., “From text to speech. The MITalk system”, Cambridge University Press, Cambridge UK, 1987.

[6] Klatt, D.H., “The Klattalk text-to-speech conversion system”, Proc. IEEE-ICASSP82, 1589-1592, 1982.

[7] Olive, J.P., “Rule synthesis of speech from dyadic units”, Proc. IEEE-ICASSP77, 568-570, 1977.

[8] Olive, J.P. 1980, “A scheme for concatenating units for speech synthesis”, Proc. IEEE-ICASSP80, 568-571.

[9] Olive, J.P. & Liberman, M., “A set of concatenative units for speech synthesis”, In: J.J. Wolff and D.H. Klatt Eds., ASA*50 Speech Communication Papers, 515-518, 1979.

[10] Sagisaka, Y., “Speech synthesis by rule using an optimal selection of nonuniform synthesis units”, Proc. IEEE-ICASSP88, 679-682, 1988.

[11] Church, K., “Stress assignment in letter to sound rules for speech synthesis”. In ACL Proceedings, 23rd Annual Meeting, pages 246–253, Morristown, NJ, 1985. Association for Computational Linguistics.1985.

[12] Campbell, W.N. & Wightman, C.W. 1992, “Prosodic coding of syntactic structure in English speech”, Proc. ICSLP92, Banff, Canada, 1167-1170.

[13] Campbell, W.N., “Synthesis units for natural English speech”, Transactions of the Institute of Electronics, Information and Communication Eng, Vol. SP 91-129, 55-62, 1992.

[14] Campbell, W.N., “CHATR: A High-Definition Speech Re-Sequencing System”, proc Eurospeech’95, Madrid/Spain, 1995.

[15] Campbell, W. N. and Black, A. W. “CHATR a multi-lingual speech re-sequencing synthesis system”. Technical Report of IEICE SP96-7, 45-52, 1996.

[16] CHATR Speech Synthesis: http://feast.his.atr.jp/chatr

[17] Iida, A., Campbell, N. and Yasumura, M. “Design and Evaluation of Synthesised Speech with Emotion”. Journal of Information Processing Society of Japan Vol. 40, 1998.

[18] Iida, A., Campbell, N., Iga, S., Higuchi, Y,. and Yasumura, Y., “A speech synthesis system with emotion for assisting communication”. In Proceedings of the ISCA Workshop on Speech and Emotion, pages 167-172, Belfast, 2000.

[19] Schroder, M., et. al., “Acoustic correlates of emotion dimensions in view of speech synthesis”, pp.87-90, In Proc Eurospeech 2001, Denmark, 2001.

[20] Ekman, P., “Universals and cuntural differences in facial expression of emotion”, in J., K. Cole (Eds), Nebraska Symposium on Motivation, pp.207-282, Lincoln, University of Nebrasaka Press, 1972.

[21] Campbell, N., “Recording Techniques for capturing natural everyday speech” pp.2029-2032, in Proc Language Resources and Evaluation Conference (LREC-02), Las Palmas, Spain, 2002

[22] Campbell, N., “Speech & Expression; the Value of a Longitudinal Corpus”, pp.183-186 in Proc Language Resources and Evaluation Conference (LREC-04), Lisbon, Portugal, 2004.

[23] Campbell, N., & Erickson, D., ”What do people hear? A study of the perception of non-verbal affective information in conversational speech”, pp. 9-28 in Journal of the Phonetic Society of Japan, V7,N4, 2004.

[24] Campbell, N., “Specifying Affect and Emotion for Expressive Speech Synthesis”, In, A. Gelbukh (Ed.) *Computational Linguistics and Intelligent Text Processing*, Proc. CICLing-2004. Lecture Notes in Computer Science, Springer-Verlag, 2004.

[25] Campbell,, N., ”Getting to the heart of the matter; Speech is more than just the Expression of Text or Language”, Keynote speech in Proc Language Resources and Evaluation Conference (LREC-04), Lisbon, Portugal, 2004.

[26] S. Deligne and F. Bimbot, “Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams”, pp.169-172 in Proc ICASSP, 1995.

[27] Campbell, N., “Listening between the lines; a study of paralinguistic information carried by tone-of-voice” pp 13-16, in Proc International Symposium on Tonal Aspects of Languages, TAL2004, Beijing, China, 2004.

[28] Campbell, N., “Extra-Semantic Protocols; Input requirements for the synthesis of dialogue speech”, pp.221-228 in Andre E., Dybkjaer, L., Minker, W., & Heisterkamp, P., (Eds) *Affective Dialogue Systems*, Springer Lecture Notes in Artificial Intelligence Series, 2004.

[29] Campbell, N., & Mokhtari, P., "Voice quality: the 4th prosodic dimension", pp.2417-2420 in Proceedings of the 15th International Congress of Phonetic Sciences (ICPhS'03), Barcelona, Spain, 2003.

[30] Campbell, N., "Perception of Affect in Speech - towards an Automatic Processing of Paralinguistic Information in Spoken Conversation", in Proc ICSLP 2004.

NICK CAMPBELL is engaged in research as a Project Leader in the Department of Emergent Communication at the ATR Network Informatics Laboratories in Kyoto, and as Research Director for the JST/CREST Expressive Speech Processing and the SCOPE 'Robot's Ears" projects. He received his PhD in Experimental Psychology from the University of Sussex in the U.K. He was invited as a Research Fellow at the IBM UK Scientific Centre, where he developed algorithms for prosody in speech synthesis, and later at the AT&T Bell Laboratories where he worked on Japanese speech synthesis. He served as Senior Linguist at the Edinburgh University Centre for Speech Technology Research before joining ATR in 1990. His research interests include non-verbal speech communication, large speech corpora, concatenative speech synthesis, and prosodic information modelling. He spends his spare time teaching postgraduate students as Visiting Professor at the Nara Institute of Science & Technology and at Kobe University in Japan. E-mail: nick@atr.jp

TRANSLATIONS AS A SEMANTIC KNOWLEDGE SOURCE

Helge Dyvik
University of Bergen (Norway)

Abstract

Theories of meaning are sometimes used to throw light on the phenomenon of translation. We argue that light can fruitfully be thrown in the opposite direction: we can use translations to get a handle on meaning. More specifically, we will motivate and present a method for the automatic extraction of wordnet-type information from translational data, and review some results. The basic insight behind the method is that much information about the semantic relations among the words in a language resides in the way in which the sets of their possible translations into some other language overlap. Therefore, if we take the translational relation between two languages as a theoretical primitive, languages can serve as each other's "semantic mirrors".

Keywords: translation, parallel corpora, wordnets, lexical semantics, thesaurus derivation.

1. Introduction

Students of translation sometimes ask what a study of meanings may teach us about translation. In this paper I want to reverse the question and ask what a study of translation may teach us about meanings.

From the perspective of the descriptive linguist, or the developer of language resources, this question seems at least as reasonable as the first one. After all, meanings appear to be far more elusive phenomena than translations: we generally feel that we know more or less what translations are, while answers tend to get much vaguer when we are asked what meanings are, or how we should distinguish them. The latter questions require theory-bound reflection, while translation is a practical task. Translations come about when translators, usually with no theoretical concern in mind, evaluate the interpretational possibilities of linguistic expressions in specific contexts, within texts with specific purposes, and then try to recreate the same interpretational possibilities in a target text serving a comparable purpose in another language. This is a normal and common kind of linguistic activity in multilingual societies – an activity which provides an empirical basis for talking about a *translational relation* between languages. Given its basis in the ubiquitous activity of practical translation, the translational relation emerges as epistemologically prior to more abstract and theory-bound notions such as 'meaning', 'synonymy' and 'inference'. What this suggests is taking the translational relation between languages as a theoretical primitive – a concept not to be defined in terms of other concepts, but assumed to be extractable from translational data by interpretive methods – and then investigate to what extent other concepts can be defined in its terms. By this move, we may hope to give semantic description more of an intersubjective basis. Besides, semantics becomes an essentially multilingual concern.

Two questions may spring to mind at this point. The first one is: Is translation really possible? and the second one: Even if it is, how can it tell us more about the semantics of each language involved than the monolingual approach? The answers, in my opinion, are: No, in a certain sense translation is impossible, and yes, precisely because perfect translation is impossible, actual translations can tell us a lot about semantics. Translation is impossible because meanings and interpretations are not like soft and pliant substances extractable from one expression in one language and mouldable without loss or modification into another expression in another language. Languages, on the contrary, are discrete structures, and meanings are entwined in the structures themselves. Therefore, during translation, things crack and snap, things disappear, and things are added, and there is hardly ever a unique correct solution to a translational task. Instead, actual translations provide a host of alternative approximations to the unattainable ideal, and this is a potential source of information: semantic insights may emerge from the way the sets of alternatives are structured. Semantic studies always depend on paraphrases, or alternative ways of saying the same thing; translations provide such alternatives from a theoretically untainted source.

There is an increasing interest in exploring the potential of translations to provide semantic insights, see e.g. Resnik and Yarowsky 1997, Ide 1999a, Ide 1999b, Diab and Resnik 2002, Ide et al. 2002, Tufis and Ion 2003, Tufis et al. 2003, Tufis et al. 2004, Priss and Old 2005. In Tufis et al. 2004 the authors combine the use of parallel corpora and aligned wordnets from different languages in order to achieve improved word sense disambiguation. In the following we will also look at the relationship between parallel corpora and wordnets, but in a different way: Rather than presupposing them as independent resources – still a rather rare luxury – we will consider to what extent one could reasonably expect to derive wordnets and similar semantic resources from parallel corpora consisting of originals aligned with their translations, based on the general ideas just sketched.

2. Semantic fields and translation

The traditional notion of a ‘semantic field’ stands for a conceptual continuum which is carved up in a certain way by a subset of the vocabulary in a given language, but which may be carved up in different ways in other languages or in different historical stages of the same language. The concept goes back to structuralist studies of lexical semantics by Jost Trier and others. The meanings of words belonging to the same semantic field are supposed to be to some extent interdependent, so that, for instance, the meaning change of a word over time has to be seen in connection with the meaning development of the words around it in the semantic field. A classical structuralist approach to the description of word meanings within a field is the use of *componential analysis*, expressed by assigning *semantic features* to the words, capturing their interrelations. This is closely related to the modern work on ontologies, in which concepts may be structured in lattices defined by feature inheritance, as in the simple example in Figure 1.

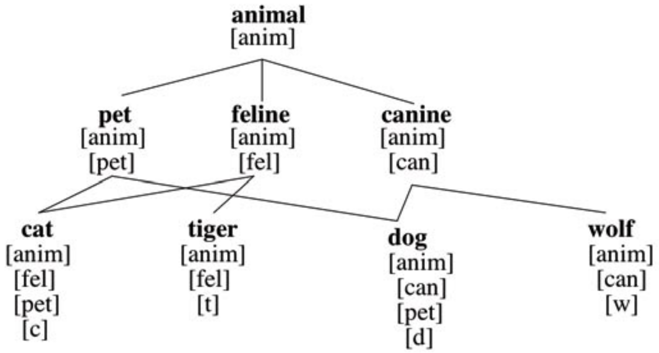


Figure 1. A simple semantic field.

In Figure 1 the most general concept is *animal*, whose single intrinsic feature [anim] is inherited by all the other concepts. One level down *pet* is a common hyperonym of *cat* and *dog*, which inherit *pet*'s intrinsic feature [pet] in addition to its inherited feature [anim], while features inherited from *feline* and *canine* similarly distinguish cats and tigers from dogs and wolves.

The point to note here is that the lattice structure can be read off from the inclusion and overlap relations among the resulting feature sets: the mother/daughter relation in the lattice is a subset/superset relation, while all nodes with intersecting feature sets are dominated by a node carrying the intersection. We should also note in passing that the intuitive semantic content of each feature is unimportant for the characterization of the lattice structure; only the distinctness and distribution of the features matter.

A difference between ontologies and semantic fields is that work on ontologies typically intends to capture constant, language-independent conceptual structures, while work on semantic fields typically intends to bring out the variability and language-specificity of the sets of terms and their interrelations: different languages may carve up the same field in different ways. Without going into the philosophical question of what the 'sameness' of semantic fields across different languages consists in, we may at least observe that the corresponding sets of terms in two languages are connected by a relation of translation. The differences between the ways in which different languages carve up the 'same' field is then reflected in the fact that this translational relation is not one-to-one; consider the classical example in Figure 2.

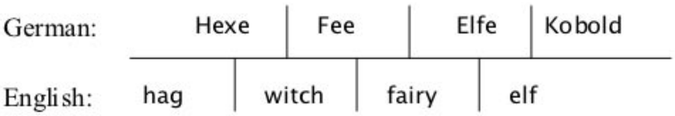


Figure 2. Different partitionings of the 'same' semantic field

In Figure 2, German *Hexe* corresponds translationally both to *hag* (an old repulsive woman, with no presupposition of magical powers) and *witch* (a woman of any age endowed with magical powers), etc. This does not imply that *Hexe* is ambiguous, only that its denotation spans the denotations of two words in the other language. Ambiguity might be involved, but that would have to be independently established; the existence of more than one translation is not enough. We may also observe how the non-transitive translational connections may tie together semantically distant words in the same semantic field: *hag* and *elf* have little to do with each other semantically, but there is a way, documented by the translational mirror image in German, of getting from the one to the other by small steps from one word to a semantically close word.

Analyses of semantic fields by means of features have also been used in a translational context; one example can be found in an article by one of the pioneers of translation theory, Eugene A. Nida (1958). Here the perspective is that of the translator faced with heavily culture-specific semantic fields of which he has scant knowledge. Hence the question is the traditional one about what a study of meanings may teach us about translation, rather than the reverse. The task is to find translational correspondences between a variety of terms for 'shaman' in two Mayan languages. The method, called 'Componential Plotting', was to make a table with the terms along one axis, and all the different functions of a shaman – healing sick, casting spells, etc. – along the other (Nida 1958:15). Then informants were asked what they would call a person performing each function, and the correspondences between terms and functions were plotted in the table. This is a nice example of an empirical semantic investigation, applied mono-lingually, leading to the assignment of semantic features (denoting shaman functions) to a set of words across two languages. A network of translational correspondences between terms in each language could then be established on the basis of shared features.

In our context this example illustrates the connection between feature sharing and translational correspondence, but we want to use that connection in order to go in the other direction – from translational correspondences to semantic features – since we are taking the translational relation as a primitive. After all, the normal case is that translation is performed without any previous, theoretically sophisticated analysis like Componential Plotting, but rather based on the existing cross-cultural competence of translators. Treating the output of translators as data is therefore not much different from treating any kind of output from language users as data for linguistic studies.

3. Translationally based representations

In reversing the direction of inference from the 'shaman' case to the case of deriving semantic features from translational data, the basic question becomes: What minimal set of semantic features, and which distribution of them, would motivate this given network of translational relations? In order to answer this question we need not consider the possible semantic interpretation of the features themselves; they are simply translationally derived formal devices whose distribution among a set of words is the only thing that matters. We may consider a simple example.

As we saw earlier, the German noun *Hexe* can be found translated into English as *hag* and *witch*; cf. Figure 3.

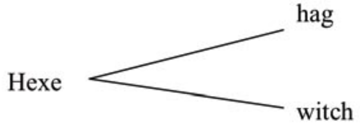


Figure 3. A simple translational correspondence

These alternative translations are obviously related to different 'aspects', or related subsenses, of the meaning of *Hexe*. The two English words indicate one way, undoubtedly among many, of dividing up the semantic potentiality of *Hexe*. In fact, we could conceive of lexical subsenses as corresponding to ordered pairs like <Hexe, hag> and <Hexe, witch> – or to sets rather than pairs, if we take several languages into account simultaneously. A translational approach to semantics sees such sets of translationally corresponding items across languages as *the primitives of semantic descriptions*. (This idea is related to the idea behind Martin Kay's 'triangulation' approach to translation.) Pairs like <Hexe, hag> can then be treated as a kind of semantic features, written [Hexe|hag] and assignable to lexical items, both to the items they were derived from (as in Figure 4), and to others, which may inherit them – a point to which we will return.

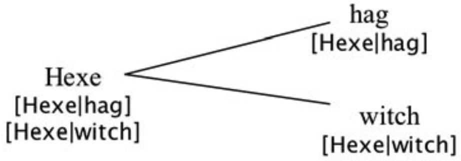


Figure 4. Assignment of translationally derived features

Intuitively, the features encode subsenses that the lexical items share with each other. In this way the features become classificatory devices, grouping lexical items together according to shared semantic properties.

4. The Semantic Mirrors method

4.1. Assumptions

Given a word-aligned parallel corpus, we may extract the set of alternative translations for each lemma in the corpus. The result is an intricate network of translational correspondences uniting the vocabularies of the two languages. This network allows us to treat each language as the 'semantic mirror' of the other, based on the ideas sketched above, in conjunction with the following assumptions:

(1) Semantically closely related words tend to have strongly overlapping sets of translations.

(2) Words with wide meanings tend to have a higher number of translations than words with narrow meanings.

(3) If a word *a* is a hyponym of a word *b* (such as *tasty* of *good*, for example), then the possible translations of *a* will probably be a subset of the possible translations of *b*.

(4) Contrastive ambiguity, i.e., ambiguity between two unrelated senses of a word, such as the two senses of the English noun *band* ('orchestra' and 'piece of tape'), tends to be a historically accidental and idiosyncratic property of individual words. Hence we don't expect to find instances of the same contrastive ambiguity replicated by other words in the language or by words in other languages. (More precisely, we should talk about ambiguous *phonological/graphic* words here, since such ambiguity is normally analysed as homonymy and hence as involving two lemmas.)

(5) Words with unrelated meanings will not share translations into another language, except in cases where the shared (graphic/phonological) word is contrastively ambiguous between the two unrelated meanings. By assumption (4) there should then be at most one such shared word.

4.2. Isolating word senses

The first step in applying the method is to use assumptions (4) and (5) to identify the set of alternative, mutually unrelated senses of each word.

We will refer to the set of translations in L2 of a word *w* in L1 as 'the first *t*-image' of *w*. Taking the first *t*-images back in L1 of all the members of *w*'s first *t*-image gives us a set of intersecting sets of words in L1; this will be referred to as *w*'s 'inverse *t*-image'. We may then make a third translational move, finding the first *t*-images in L2 of all the members of the union of *w*'s inverse *t*-image; this gives us a set of intersecting sets of words in L2, which we will call *w*'s 'second *t*-image'.

We may exemplify sense individuation by means of *t*-images with a corpus example taken from The English-Norwegian Parallel Corpus (ENPC), a corpus which comprises approximately 2.6 million words, originals and translations included. The corpus contains fiction as well as non-fiction and English originals translated into Norwegian as well as the other way around (Johansson et al. 1996). The example[1] is based on manual word alignment. Figure 5 shows the contrastively ambiguous Norwegian noun *rett* (which can mean, i.a., 'dish' and 'court of law') with its first and inverse *t*-images. Obviously, *rett* vill be a member of all the sets in its inverse *t*-image, but this is not shown in the figure, to keep it reasonably simple. However, it should be kept in mind, since it means that all the sets that are shown as intersecting in the inverse *t*-image actually contain *rett* as well in their intersections. This is crucial, given assumption (5) above, because it means that all the intersections contain at least two members, which in turn means that the sets are assumed to contain semantically related words. With only one word in the intersection, the chances are that this word may be contrastively ambiguous between the senses represented by each set. For example, this is the case with the first *t*-images of *law* and *food*, which only contain *rett* itself in the intersection (not shown in the figure). Given that the two *t*-images are not indirectly connected by means of intersections with other sets, this leads to the conclusion that *rett* is contrastively ambiguous between a *law* sense and a *food* sense.

To put it more carefully: The sets in the inverse *t*-image are divided in groups based on intersections containing words in addition to *rett* itself, and each such group is assumed to correspond to a distinct sense of *rett*. Mapping these groups back on the first *t*-image gives a partitioning of it into *sense partitions*, indicated by horizontal lines in Figure 5. Thus we individuate four senses *rett1, rett2, rett3* and *rett4*, each associated with its own first *t*-image.

While the result looks plausible as far as the separation of the *food* and *law* senses is concerned, it also illustrates the inevitable limitations of using a finite corpus: *course* really belongs in the *food* partition, but constitutes its own spurious sense here because the corpus happens not to contain any translations of *course*, apart from *rett*, shared with any of the other food-related English words.

[1] The example is taken from Lyse (2003).

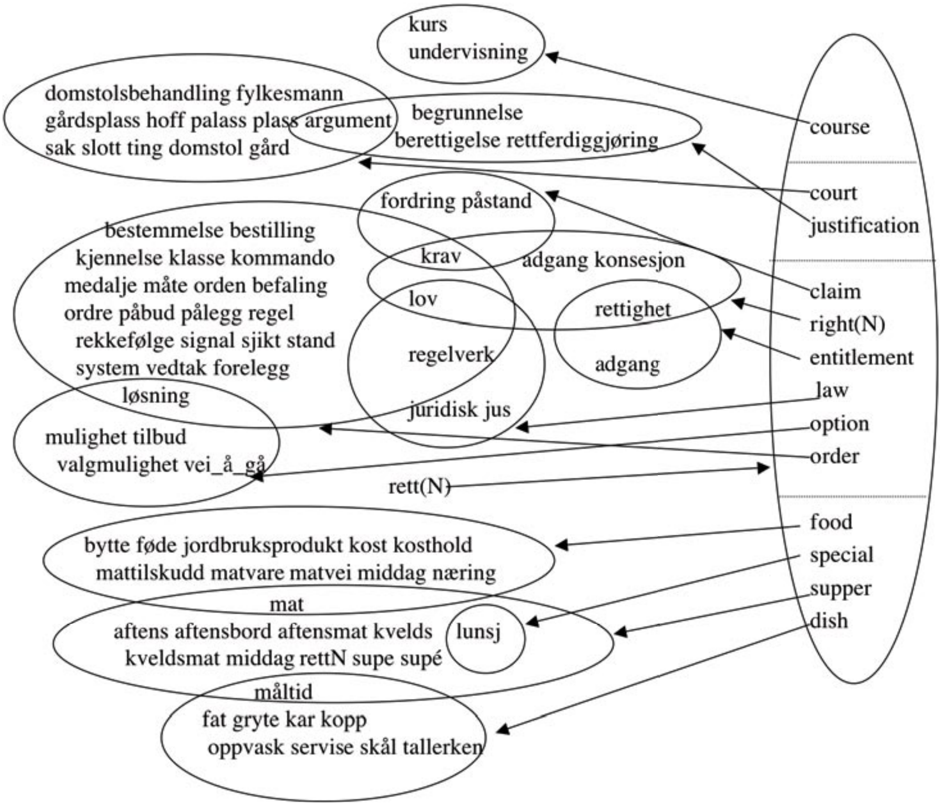

kurs
undervisning
domstolsbehandling fylkesmann
gårdsplass hoff palass plass argument
sak slott ting domstol gård
begrunnelse
berettigelse rettferdiggjøring
fordring påstand
bestemmelse bestilling
kjennelse klasse kommando
medalje måte orden befaling
ordre påbud pålegg regel
rekkefølge signal sjikt stand
system vedtak forelegg
krav
adgang konsesjon
lov
regelverk
juridisk jus
rettighet
adgang
løsning
mulighet tilbud
valgmulighet vei_å_gå
rett(N)
bytte føde jordbruksprodukt kost kosthold
mattilskudd matvare matvei middag næring
mat
aftens aftensbord aftensmat kvelds
kveldsmat middag rettN supe supé
lunsj
måltid
fat gryte kar kopp
oppvask servise skål tallerken
course
court
justification
claim
right(N)
entitlement
law
option
order
food
special
supper
dish

Figure 5. The first (on the right) and inverse (on the left) *t*-images of the noun *rett*

4.3. Semantic fields and feature assignment

Once senses are individuated in the manner described in both languages, they can be grouped into *semantic fields*. In our translational approach, the semantic fields are isolated on the basis of overlapping *t*-images: two senses belong to the same semantic field if they have intersecting first *t*-images (after sense individuation one member in the intersection is sufficient), or if there is a sequence of such intersecting *t*-images joining them.

We treat translational correspondence as a symmetric relation (disregarding the direction of translation), and as a consequence we get paired semantic fields in the two languages involved. Each field f1 and f2 in such a pair imposes a subset structure on the other, since all the *t*-images of the members of f1 will be subsets of f2, and *vice versa*. By assumptions (1-3) above, rich information about the semantic relations among the senses can be derived from this subset structure.

Taking the food-related sense of *rett* (*rett4*) as a starting point, we can collect its semantic field by finding all other senses with directly or indirectly intersecting *t*-images. This is shown on the left in Figure 6. The corresponding field in English is shown on the right. Furthermore, the subset structures imposed by the *t*-images are also indicated.

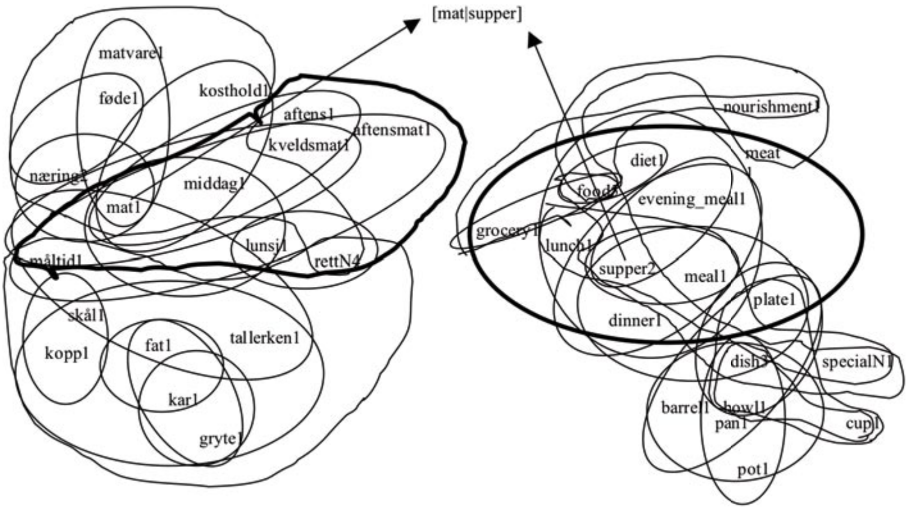


Figure 6. Paired semantic fields from Norwegian and English

The fact that a sense is a member of many subsets, i.e., of many *t*-images, indicates that it has many translational partners in the other field. By assumption (2) such senses are expected to have wide meanings as compared to other senses in the field. As expected, senses such as *food5* and *mat1* ('food') in Figure 6 constitute such peaks in the subset structures (although *supper2* happens to outrank *food5* in the English field, being a member of an even higher number of subsets). Furthermore, the fact that two senses are co-members of many subsets means that they share many translations and hence ought to be closely related semantically.

In this way the subset structures contain rich information about the semantic relations among the senses, and the next step is to encode this information in feature sets associated with the senses. The procedure[2] is to start from the 'peaks', i.e., from the pair of senses that are both translationally related and members of the highest number of subsets – *mat1* and *supper2* in the example. A feature is constructed from these two senses, as also illustrated in Figure 6. The feature is assigned to the two senses *mat1* and *supper2*, and is then inherited by 'lower' senses, i.e., by all senses ranked lower than *mat1* within the first *t*-image of *supper2*, and by all senses ranked lower than *supper2* within the first *t*-image of *mat1*. The *t*-images in question are marked by bold lines in Figure 6. Then the procedure moves on iteratively to the next highest peaks – *middag1* ('dinner') and *food5* in the example – constructing the feature [middag1|food5] and assigning it according to the same principles. The final result is feature sets assigned to all the senses in the two fields. By hypothesis, feature set inclusion now expresses a hyperonymy/hyponymy relation, e.g. as in the two senses *food5* : *lunch1*:

food5	**lunch1**
[mat1|supper2]	[mat1|supper2]
[middag1|food5]	[middag1|food5]
	[lunsj1|meal1]
	[lunch1]

[2] The procedure is described in more detail in Dyvik 1998:80ff.

The full set of senses in a field is thus partially ordered by set inclusion. We can construct an upper semilattice from this set, allowing us to compare the distances between all the senses in the field. An upper semilattice is a partially ordered set in which each pair of elements has a least upper bound. Applied to our case this means that for each pair of feature sets, either one set includes the other or there is a third feature set consisting of the intersection of the two sets. By adding elements with such intersections whenever they don't exist already, we construct an upper semilattice from a semantic field. Intuitively, the added elements are 'virtual hyperonyms' of the intersecting elements – potential senses that happen not to be lexicalized in the language (or at least not to occur in the corpus). We label the added elements as indexed X'es. Thus, given two intersecting sets such as the sets for *busy2* and *alive2* in Figure 7, we construct the node *X1* carrying the intersection of the feature sets:

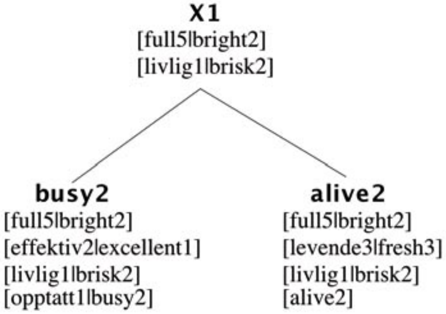


Figure 7. Adding X-nodes to construct a semilattice.

Figure 8 shows a small part of a semilattice for adjectives, based on manual word alignment of the ENPC, in which the adjective *brilliant* unites senses related to cleverness and radiance.

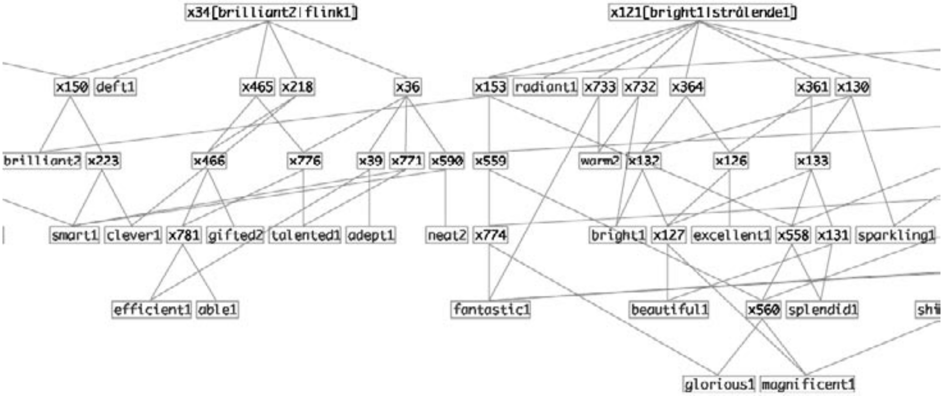


Figure 8. Part of a semilattice for adjectives

4.4. Deriving thesaurus entries

The feature lattices contain some of the information represented in thesaurus entries, and we may derive rudimentary thesaurus-like entries from them. Derivation of a thesaurus entry for a sense *s* involves collecting senses that are sufficiently related to *s* from the semilattice, and sort them into hyperonyms, hyponyms and synonyms of *s*. Basically, a related sense of *s* is a sense sharing features (and hence translations) with *s*. A hyperonym of *s* is then a sense *h* from which *s* has inherited a feature, provided that the number of senses having inherited this feature exceeds a certain threshold (called *SynsetLimit*); the latter provision ensures that hyperonyms have sufficiently wide meanings. Hyponyms of *s* are, conversely, senses which have inherited an inherent feature of *s*, with the same provision about the number of heirs. Synonyms and 'related words' are also identified on the basis of certain kinds of feature sharing.

Furthermore the sense *s* can be divided into mutually related subsenses. Each feature assigned to *s* potentially represents a distinct subsense; whether two features *f1* and *f2* should be considered as belonging to the same subsense or not, can be determined on the basis of the sets of senses to which *f1* and *f2* are assigned. If the intersection of these sets of senses exceeds a certain theshold (called *OverlapThreshold*), the features are not considered as representing distinct subsenses.

For example, with a certain setting of the thresholds the following entry is derived for one sense of the adjective *brilliant*:

brilliant
Hyperonyms: bright‹1›.
Subsense (i)
(**Translation:** skarp, flink.)
Synonyms: able, adept, clever, deft‹1›, efficient‹1›, fierce, gifted‹2›, neat‹2›, smart‹1›, talented‹1›.
Related words: burning‹1›, harsh‹1›, hot‹1›, keen‹1›, piercing‹2›, sharp‹1›, shrill‹1›, spiny‹1›, stark‹1›, steep‹1›, stinging‹1›.
Subsense (ii)
(**Translation:** fantastisk, strålende.)
Synonyms: amazing‹1›, enormous‹1›, exceptional, extraordinary‹1›, fantastic, glorious‹1›, magnificent‹1›, marvellous, remarkable‹1›, spectacular‹1›, splendid‹1›, startling‹1›, surprising‹1›, unusual‹1›.

Increasing the *OverlapThreshold* leads to a splitting up of Subsense (i) in two subsenses, separating the 'hot' and 'sharp' aspect of the sense from the 'clever' and 'efficient' aspect:

brilliant
Hyperonyms: bright‹1›.
Subsense (i)
(**Translation:** flink.)
Synonyms: able, adept, clever, deft‹1›, efficient‹1›, gifted‹2›, neat‹2›, smart‹1›, talented‹1›.
Subsense (ii)
(**Translation:** fantastisk.)
Synonyms: amazing‹1›, enormous‹1›, exceptional, extraordinary‹1›, fantastic, glorious‹1›, magnificent‹1›, marvellous, remarkable‹1›, spectacular‹1›, splendid‹1›, startling‹1›, surprising‹1›, unusual‹1›.
Subsense (iii)
(**Translation:** skarp.)
Synonyms: fierce.
Related words: burning‹1›, clever, harsh‹1›, hot‹1›, keen‹1›, piercing‹2›, sharp‹1›, shrill‹1›, smart‹1›, spiny‹1›, stark‹1›, steep‹1›, stinging‹1›.

5. Empirical findings

The Semantic Mirrors method has been explored in a project involving automatic word alignment[3] of the ENPC and comparison of results from manually aligned and automatically aligned data, comparison of the output of our method with existing resources such as the Princeton Wordnet and Merriam-Webster's Thesaurus[4], and testing of the method as a basis for word sense disambiguation[5] (presently only with preliminary results). We may briefly summarize some of our findings so far as follows:

- The method is vulnerable to the increased noise introduced by automatic word alignment: precision and recall in the thesaurus output from automatically aligned data as compared with the output from manually aligned data seems to be lower than the precision and recall of the automatic word alignment itself as compared with manual word alignment.
- It is hard to find a suitable gold standard for the evaluation of the thesaurus output. When using Merriam-Webster's Thesaurus or Princeton Wordnet as gold standards for the sets of semantically related words associated with the thesaurus entries, precision and recall is low, but not very much worse than the results obtained when we compare the established resources Merriam-Webster and Princeton Wordnet with each other.
- There is a distinct difference between different parts of speech: the method gives better results for adjectives than for nouns and verbs, and abstract nouns give better results than concrete nouns. With concrete nouns very few hyperonym/hyponym-relations are discovered, probably because translational relations between hyperonyms and hyponyms are more rare with concrete nouns than with, e.g. adjectives: translating *dog* with a word meaning 'animal' doesn't happen as often as translating *tasty* with a word meaning 'good', for example. Besides, adjectives, typically denoting single properties, tend to form tighter groups of closely related members than nouns, which typically denote clusters of properties; this may explain why adjectives tend to have more alternative translations than nouns.

In sum, if high-quality translational data can be provided, the method clearly seems to provide some useful results.

6. References

Diab, Mona & Philip Resnik 2002. An Unsupervised Method for Word Sense Tagging using Parallel Corpora. *40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, July, 2002.

Dyvik, Helge 1998a. A translational basis for semantics. In Stig Johansson and Signe Oksefjell (eds.): *Corpora and Crosslinguistic Research: Theory, Method, and Case Studies*. Amsterdam: Rodopi 1998, 51-86.

Dyvik, Helge 1998b. Translations as semantic mirrors. In *Proceedings of Workshop W13: Multilinguality in the lexicon II*, pp. 24.44, Brighton, UK. The 13th biennial European Conference on Artificial Intelligence ECAI 98.

[3] The algorithm for automatic word alignment was developed by Sindre Sørensen.

[4] This evaluation is carried out by Martha Thunes.

[5] Word sense disambiguation is explored by Gunn Inger Lyse.

Dyvik, Helge 2002. Translations as semantic mirrors: from parallel corpus to wordnet. *Language and Computers*, 1 April 2004, vol. 49, iss. 1, 311-326(16) Rodopi.

Ide, Nancy 1999. Word sense disambiguation using cross-lingual information. In *Proceedings of ACH-ALLC '99 International Humanities Computing Conference*, Charlottesville, Virginia. http://jefferson.village.virginia.edu/ach-allc.99/proceedings

Ide, Nancy 1999. Parallel translations as sense discriminators. In *SIGLEX99: Standardizing Lexical Resources, ACL99 Workshop*, College Park, Maryland, 52-61.

Ide, Nancy, Tomas Erjavec & Dan Tufis 2002. Sense Discrimination with Parallel Corpora. In *Proceedings of ACL'02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Philadelphia, 54-60.

Johansson, Stig, Jarle Ebeling, and Knut Hofland 1996. Coding and aligning the English-Norwegian Parallel Corpus. In K. Aijmer, B. Altenberg, and M. Johansson (eds.) 1996. *Languages in contrast. Papers from a symposium on text-based cross-linguistic studies in Lund, 4-5 March 1994*, 73-85. Lund: Lund University Press, 87-112.

Lyse, Gunn Inger 2003. *Fra speilmetoden til automatisk ekstrahering av et betydningstagget korpus for WSD-formål.* Masters thesis, University of Bergen.

Nida, Eugene A. 1958. Analysis of Meaning and Dictionary Making. In *Language Structure and Translation. Essays by Eugene A. Nida*. Selected and Introduced by Anwar S. Dil, Stanford University Press 1975, 1-23.

Priss, Uta and John Old 2005. Conceptual Exploration of Semantic Mirrors. In Ganter, Godin (eds.) *Formal Concept Analysis: Third International Conference, ICFCA 2005*, Springer Verlag.

Tufis, Dan and Radu Ion 2003. Word sense clustering based on translation equivalence in parallel texts; a case study in Romanian. In *Proceedings of the International Conference on Speech and Dialog – SPED*, 13-26, Bucharest.

Tufis, Dan, Radu Ion, Nancy Ide 2004. Fine-Grained Word Sense Disambiguation Based on Parallel Corpora, Word Alignment, Word Clustering and Aligned Wordnets. In *Proceedings of the 20th International Conference on Computational Linguistics, COLING2004*, 1312-1318, Geneva.

Resnik, Philip Stuart & David Yarowsky 1997. A perspective on word sense disambiguation methods and their evaluation, position paper presented at the ACL SIGLEX Workshop on *Tagging Text with Lexical Semantics: Why, What, and How?*, held April 4-5, 1997 in Washington, D.C., USA in conjunction with ANLP-97.

HELGE DYVIK is professor of general linguistics at the University of Bergen, Norway. He received his dr.philos. degree in 1981 based on a study of syntactic theory and linguistic methodology. His present research interests comprise semantics and translation, machine translation, and the development of computational grammars. He has previously published studies on Old Norse, Old English and Vietnamese, and on the interpretation of runic inscriptions. He is presently head of the programme committee for the research programme on language technology under The Nordic Council of Ministers.

SPONTANEOUS SPEECH RECOGNITION AND SUMMARIZATION

Sadaoki Furui
Tokyo Institute of Technology (Japan)

Abstract

This paper overviews recent progress in the development of corpus-based spontaneous speech recognition technology focusing on various achievements of a Japanese 5-year national project "Spontaneous Speech: Corpus and Processing Technology". Although speech is in almost any situation spontaneous, recognition of spontaneous speech is an area which has only recently emerged in the field of automatic speech recognition. Broadening the application of speech recognition depends crucially on raising recognition performance for spontaneous speech. For this purpose, it is necessary to build large spontaneous speech corpora for constructing acoustic and language models. Because of various spontaneous-speech specific phenomena, recognition of spontaneous speech requires various new techniques. These new techniques include flexible acoustic modeling, sentence boundary detection, pronunciation modeling, acoustic as well as language model adaptation, and automatic summarization. Particularly automatic summarization including indexing, a process which extracts important and reliable parts of the automatic transcription, is expected to play an important role in building various speech archives, speech-based information retrieval systems, and human-computer dialogue systems.

Keywords: spontaneous speech, corpus, recognition, model adaptation, summarization

1. Introduction

Read speech and similar types of speech, e.g. news broadcasts reading a text, can be recognized with accuracy higher than 95%, using the state-of-the-art speech recognition technology. However, recognition accuracy drastically decreases for spontaneous speech (Furui, 2003a; Furui, 2003b). One of the major reasons for this decrease is that acoustic and language models used up until now have generally been built using written language or speech read from a text. Spontaneous speech and speech from written language are very different, both acoustically and linguistically. Spontaneous speech includes filled pauses, repairs, hesitations, repetitions, partial words, and disfluencies. It is quite interesting to note that, although speech is almost always spontaneous, spontaneous speech recognition is a special area that emerged only about 10 years ago within the wider field of automatic speech recognition (e.g. Shinozaki et al., 2001;

Sankar et al., 2002; Gauvain and Lamel; 2003; Evermann et al., 2004; Schwartz et al., 2004). Broadening the application of speech recognition depends crucially on raising the recognition performance for spontaneous speech.

In order to increase recognition performance for spontaneous speech, it is necessary to build acoustic and language models for spontaneous speech. Current methods applying statistical language modeling such as bigrams and trigrams of words or morphemes to spontaneous speech corpus may prove to be inadequate. Our knowledge of the structure of spontaneous speech is currently insufficient to achieve the necessary breakthroughs. Although spontaneous speech effects are quite common in human communication and may increase in human machine discourse as people become more comfortable conversing with machines, modeling of speech disfluencies is only in the initial stage. Since spontaneous speech includes various redundant expressions, recognition of spontaneous speech will require a paradigm shift from simply recognizing speech, where transcribing the spoken words is the primary focus, to understanding where underlying messages of the speaker are extracted.

2. Categories of speech recognition tasks

Speech recognition tasks can be classified into four categories, as shown in Table 1, according to two criteria: whether it is targeting utterances from human to human or human to computer, and whether the utterances have a dialogue or monologue style (Furui, 2003b). The table lists typical tasks for each category.

Table 1. Categorization of speech recognition tasks

	Dialogue	Monologue
Human to human	(Category I) Switchboard, Call Home (Hub 5), meeting, interview	(Category II) Broadcast news (Hub 4), other programs, lecture, presentation, voice mail
Human to machine	(Category III) ATIS, Communicator, information retrieval, reservation	(Category IV) Dictation

Category I targets human-to-human dialogues and includes DARPA-sponsored Switchboard and Call Home (Hub 5) tasks. Speech recognition research in this category aiming to make minutes of meetings (e.g. Janin et al., 2004) has recently started. Waibel et al. have been investigating a meeting browser that observes and tracks meetings for later review and summarization (Waibel and Rogina, 2003). Akita et al. have investigated techniques for archiving discussions (Akita et al., 2003). In their method, speakers are automatically indexed in an unsupervised way, and speech recognition is performed using the results of indexing. Processing human-human conversational speech under unpredictable recording conditions and vocabularies presents new challenges for spoken language processing.

A relatively new task classified into this category is the MALACH (Multilingual Access to Large spoken ArCHives) project (Oard, 2004). Its goal is to advance the state-of-the-art technology for access to large multilingual collections of spontaneous conversational speech by exploiting an unmatched collection assembled by the Survivors of the Shoah Visual History Foundation (VHF). This collection is indeed a challenging task because of heavily accented, emotional and elderly spontaneous characteristics. Named entity tagging, topic segmentation, and unsupervised topic classification are also being investigated.

Tasks belonging to Category II, which targets recognizing human-to-human monologues, include transcription of broadcast news (Hub 4), news programs, lectures, presentations, and voice mails (e.g. Hirschberg et al., 2001). Speech recognition research in this category has recently become very active. Since the utterances in the Category II are made with the expectation that the audience can correctly understand what is spoken in the one-way communication, they are relatively easier to recognize than the utterances in Category I. If high recognition performance is achieved, a wide range of applications, such as making lecture notes, records of presentations and closed captions, archiving and retrieving these records, and retrieving voice mails, will be realized.

Most of the practical application systems widely used now are classified as Category III, recognizing the utterances in human-computer dialogues, such as in airline information services tasks. DARPA-sponsored projects including ATIS and Communicator have laid the foundations of these systems. Unlike other categories, the systems in the Category III are usually designed and developed after clearly defining the application/task. The machines that we have attempted to design so far are, almost without exception, limited to the simple task of converting a speech signal into a word sequence and then determining, from the word sequence, a meaning that is “understandable”. Here, the set of understandable messages is finite in number, each being associated with a particular action (e. g., route a call to a proper destination or issue a buy order for a particular stock). In this limited sense of speech communication, the focus is detection and recognition rather than inference and generation.

Various research has made clear that the utterances spoken by people talking to computers, such as those in Categories III and IV, especially when the speaker is conscious, are acoustically as well as linguistically very different from those spoken to other people, such as those in Categories I and II. One of the typical tasks belonging to Category IV, which targets the recognition of monologues performed when people are talking to a computer, is dictation. Various commercial software for such purposes have been developed. Since the utterances in Category IV are made with the expectation that the utterances will be converted exactly into texts with correct characters, their spontaneity is much lower that that in Category III. In the four categories, spontaneity is considered to be the highest in Category I and the lowest in Category IV.

3. Spontaneous speech corpora

3.1. Issues of corpus construction

The appetite of today's statistical speech processing techniques for training material are well described by the aphorism: "There's no data like more data." Large structured collections of speech and text are essential for progress in speech recognition research. Unlike the traditional approach, in which knowledge of speech behavior is "discovered" and "documented" by human experts, statistical methods provide an automatic procedure to directly "learn" regularities in the speech data. The need for a large set of good training data is, thus, more critical than ever. However, establishing a good speech database for the computer to uncover the characteristics of the signal is not a straightforward process. There are basically two broad issues to be carefully considered: one being the content and its annotation, and the other the collecting mechanism.

The recorded data needs to be verified, labeled, and annotated by people whose knowledge is introduced into the design of the system through its learning process (i.e. via supervised training of the system after the data has been labeled). Labeling and annotation for spontaneous speech can easily become unmanageable. For example: how do we annotate speech repairs and partial words? how do the phonetic transcribers reach a consensus in acoustic-phonetic labels when there is ambiguity? and how do we represent a semantic notion? Errors in labeling and annotation will result in system performance degradation. How to ensure the quality of the annotated results is thus a major concern. Research limited only to automating or creating tools to assist the verification procedure is in itself an interesting subject.

3.2. Corpus of Spontaneous Japanese (CSJ)

In the above-mentioned context, a 5-year Science and Technology Agency Priority Program entitled "Spontaneous Speech: Corpus and Processing Technology" was conducted in Japan from 1999 to 2004 (Furui, 2003a), and a large-scale spontaneous speech corpus, Corpus of Spontaneous Japanese (CSJ), consisting of roughly 7M words with a total speech length of 650 hours was built (Maekawa, 2003; Maekawa et al., 2004).

Mainly recorded are monologues such as academic presentations (AP) and extemporaneous presentations (EP). AP is live recordings of academic presentations in nine different academic societies covering the fields of engineering, social science and humanities. EP is studio recording of paid layman speakers' speech on everyday topics like "the most delightful memory of my life" presented in front of a small audience and in a relatively relaxed atmosphere. The age and gender of EP speakers are more balanced than that of AP speakers. The CSJ also includes some dialogue speech for the purpose of comparison with monologue speech. The recordings were manually given orthographic and phonetic transcription. Spontaneous speech-specific phenomena, such as filled pauses, word fragments, reduced articulation and mispronunciation, as well as non-speech events like laughter and coughing were also carefully tagged.

One-tenth of the utterances, hereafter referred to as the Core, were tagged manually and used for training a morphological analysis and part-of-speech (POS) tagging program (Uchimoto et al., 2003) for automatically analyzing all of the 650-hour utterances (see Fig. 1). The Core consists of 70 APs, 107 EPs, 18 dialogues and 6 read speech files. They were also tagged with para-linguistic/intonation information, dependency-structure, discourse structure, and summarization. For intonation labeling of spontaneous speech, the traditional J_ToBI (Venditti, 1997) was extended to X_JToBI (Maekawa et al., 2002), in which inventories of tonal events as well as break indices were considerably enriched.

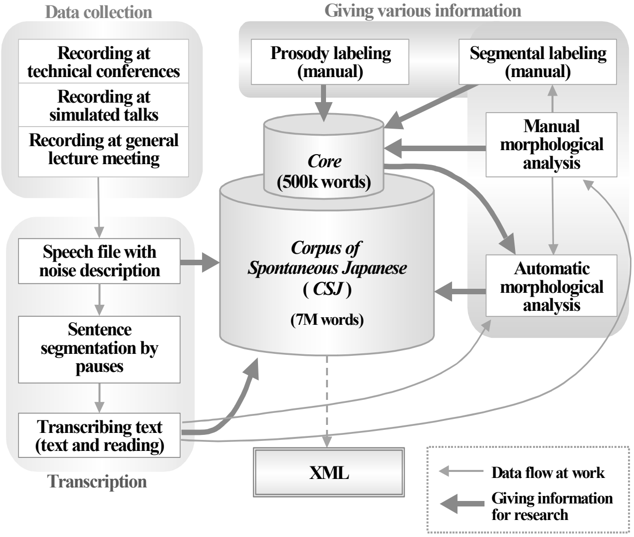

Data collection
Giving various information
Recording at technical conferences
Recording at simulated talks
Recording at general lecture meeting
Prosody labeling (manual)
Segmental labeling (manual)
Core (500k words)
Manual morphological analysis
Speech file with noise description
Corpus of Spontaneous Japanese (CSJ) (7M words)
Automatic morphological analysis
Sentence segmentation by pauses
Transcribing text (text and reading)
Transcription
XML
Data flow at work
Giving information for research

Fig. 1. CSJ corpus construction

4. Progress made in spontaneous speech recognition using the CSJ

4.1. Effectiveness of the CSJ

By constructing acoustic and language models using the CSJ, recognition errors for spontaneous presentation were reduced to roughly half compared to models constructed using read speech and written text (Furui, 2003a; Shinozaki et al., 2001). Increasing the size of training data for acoustic and language models has significantly decreased the recognition error rate (WER: word error rate), and the best WER of 25.3% was obtained when the whole training data set (510 hours, 6.84M words) was used (Ichiba et al.,2004; Furui et al., 2005). When the acoustic model was constructed using the whole training data set and the language model training data size was increased from 1/8 (0.86M words) to 8/8 (6.84M words), the WER, the perplexity and the OOV were relatively reduced by 17%, 19%, and 62%, respectively. On the other hand, when the language model was made using the whole training data set (6.84M words) and the acoustic model training data was increased from 1/8 (68 hours) to 8/8 (510 hours), the WER was reduced by 6.3%. The WER was almost saturated by using the whole data set.

4.2. Pronunciation variation

In spontaneous speech, pronunciation variation is so diverse that multiple surface form entities are needed for many lexical items. Kawahara et al. (Kawahara et al., 2004) have found that statistical modeling of pronunciation variations integrated with language modeling is effective in suppressing false matching of less frequent entries. They have adopted a trigram model of word-pronunciation entries. Since both orthographic and phonetic transcriptions of the CSJ were made manually for each unit of utterance (sentence), word-based automatic alignment between them was performed to obtain the pronunciation entries for each word. This was incorporated as a post-processor of the morphological analyzer. Heuristic thresholding was applied to eliminate erroneous patterns, in which pronunciation entries whose occurrence probability in each lexical item is lower than a threshold were eliminated. As a result, 30,820 word-pronunciation entries (24,437 distinct words) were obtained, on which a trigram model was trained. Experimental results show that the word-pronunciation trigram model is more effective than simply adding the pronunciation probability in the decoding process.

4.3. Sentence boundary detection

Another difficulty of spontaneous speech recognition is that generally no explicit sentence boundary is given. Therefore, it is impossible to recognize spontaneous speech sentence by sentence. Kawahara et al. developed a decoder in which no sentence boundaries are required (Kawahara et al., 2001). The decoder can handle very long speech with no prior sentence segmentation. Experimental results show that the new decoder performed better than the previous version using sentence boundaries. Based on transcription results and pause lengths, sentence boundaries are automatically determined and punctuation marks are given. Specifically, a linguistic likelihood ratio between a model including sentence boundary and a model without boundary is compared with a threshold and then a decision is made.

4.4. Acoustic model adaptation

Word accuracy varies largely from speaker to speaker. There exist many factors that affect the accuracy of spontaneous speech recognition. They include individual voice characteristics, speaking manners, styles of language (grammar), vocabularies, topics, and noise, such as coughs. Even if utterances are recorded using the same close-talking microphones, acoustic conditions still vary according to the recording environment. A batch-type unsupervised adaptation method has been incorporated to cope with speech variation in the CSJ utterances (Shinozaki et al., 2001). The MLLR method using a binary regression class tree to transform Gaussian mean vectors was employed. The regression class tree was made using a centroid-splitting algorithm.

The actual classes used for transformation were determined at run time according to the amount of data assigned to each class. By applying the adaptation, the error rate was reduced by 15% relative to the speaker independent condition.

4.5. Language model adaptation

Lussier et al. investigated combinations of unsupervised language model adaptation methods for CSJ utterances (Lussier et al., 2004). Data sparsity is a common problem shared by all speech recognition tasks but it is especially acute in the case of spontaneous speech recognition. The method proposed combines information from two readily available sources, clusters of presentations from the training corpus and the transcription hypothesis, to create word-class n-gram models that are then interpolated with a general language model. The interpolation coefficient is estimated based on EM algorithm using a development set. Since this method performs in an offline manner using whole recognition results to suppress influences of local recognition errors, it is more robust against recognition errors than online adaptation methods. Experimental results show that a relative reduction in word error rate of 5-10% is obtained on the CSJ test sets.

4.6. Massively Parallel Decoder-based recognition

Shinozaki et al. have proposed using a combination of cluster-based language models and acoustic models in the framework of a Massively Parallel Decoder (MPD) to cope with the problem of acoustic as well as linguistic variations of presentation utterances (Shinozaki and Furui, 2004). MPD is a parallel decoder that has a large number of decoding units, in which each unit is assigned to each combination of element models. Likelihood values produced by all the decoding units are compared, and the hypothesis having the largest likelihood is selected as the recognition result. The system runs efficiently on a parallel computer, and thus the turnaround time is comparable to the conventional decoder using a single model and processor. In experiments conducted using presentation speeches from the CSJ, two types of cluster models have been investigated: presentation-based cluster models and utterance-based cluster models. It has been confirmed that utterance-based cluster models give significantly lower recognition error rate than presentation-based cluster models in both language and acoustic modeling. It has also been shown that roughly 100 decoding units are sufficient in terms of recognition rate; and, in the best setting, 12% reduction in word error rate was obtained in comparison with the conventional decoder.

5. Spontaneous speech summarization

Spontaneous speech is ill-formed and very different from written text. Spontaneous speech usually includes redundant information such as disfluencies, fillers, repetitions, repairs and word fragments. In addition, irrelevant information caused by recognition errors is usually inevitably included when spontaneous speech is transcribed. Therefore, an approach in which all words are simply transcribed is not an effective one for spontaneous speech. Instead, speech summarization which extracts important information and removes redundant and incorrect information is ideal for recognizing spontaneous speech. Speech summarization is also expected to reduce time needed for reviewing speech documents and improve the efficiency of document retrieval.

Speech summarization has a number of significant challenges that distinguish it from general text summarization. Applying text-based technologies (Mani and Maybury, 1999) to speech is not always workable and often they are not equipped to capture speech specific phenomena (Kolluru et al, 2003; Christensen et al., 2003). One fundamental problem with the speech summarization is that they contain speech recognition errors and disfluencies. We have proposed a two-stage summarization method consisting of important sentence extraction and word-based sentence compaction, as shown in Fig. 2 (Hori and Furui, 2003; Hori et al., 2003; Kikuchi et al., 2003). After removing all the fillers based on speech recognition results, a set of relatively important sentences is extracted, and sentence compaction is applied to the set of extracted sentences. The ratio of sentence extraction and compaction is controlled according to a summarization ratio initially determined by the user. Sentence and word units are extracted from the speech recognition results and concatenated for producing summaries so that they maximize the weighted sum of linguistic likelihood, amount of information, confidence measure, and grammatical likelihood of concatenated units. The proposed method has been applied to summarization of broadcast news utterances as well as unrestricted-domain spontaneous presentations and has been evaluated by objective and subjective measures. It has been confirmed that the proposed method is effective in both English and Japanese speech summarization.

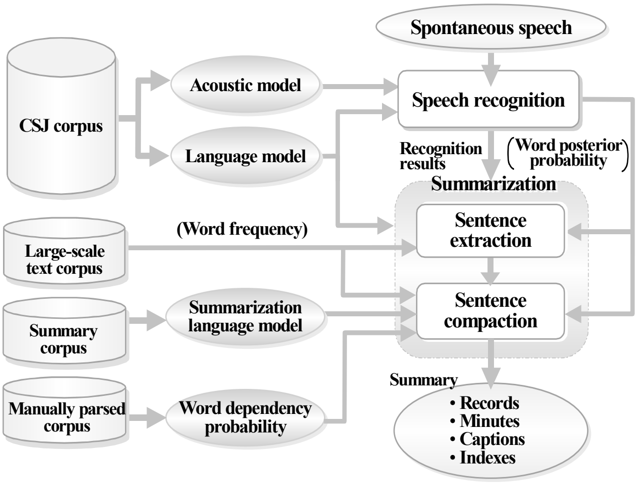


Fig. 2. A two-stage automatic speech summarization system

Speech summarization technology can be applied to any kind of speech document and is expected to play an important role in building various speech archives including broadcast news, lectures, presentations, and interviews. Summarization and question answering (QA) perform similar tasks, in that they both map an abundance of information to a (much) smaller piece which is then presented to the user. Therefore, speech summarization research will help the advancement of QA systems using speech documents. By condensing important points of long presentations and lectures and presenting them in a summary speech, the system can provide the listener with a valuable means for absorbing more information in a much shorter time.

6. Conclusions and future research

Since speech is the most natural and effective method of communication between human beings, various important speech documents, including lectures, presentations, meeting records and broadcast news, are produced everyday. However, it is not easy to quickly review, retrieve, selectively disseminate, and reuse these speech documents, if they are simply recorded as audio signal. Therefore, automatically transcribing speech using speech recognition technology is a crucial aspect of creating knowledge resources from speech.

Speech recognition technology is expected to be applicable not only to indexing of speech data (lectures, broadcast news, etc.) for information extraction and retrieval, but also to closed captioning and aids for the handicapped. Broadening these applications depends crucially on raising the recognition performance of spontaneous speech. Various spontaneous speech corpora and processing technologies have recently been created under several recent projects. However, how to incorporate filled pauses, repairs, hesitations, repetitions, partial words, and disfluencies still poses a big challenge in spontaneous speech recognition.

The large-scale spontaneous speech corpus, CSJ (Corpus of Spontaneous Japanese) will be stored with XML format in a large-scale database system developed by the COE (Center of Excellence) program “Framework for Systematization and Application of Large-scale Knowledge Resources” at Tokyo Institute of Technology so that the general population can easily access and use it for research purposes (Furui, 2004). Since the recognition accuracy for spontaneous speech is still rather low, the collection of the corpus will be continued in the COE program in order to increase coverage of variations in spontaneous speech.

Spectral analysis using various styles of utterances in the CSJ shows that the spectral distribution/difference of phonemes is significantly reduced in spontaneous speech compared to read speech (Furui et al., 2005). This is considered as one of the reasons for the difficulty of spontaneous speech recognition. It has also been observed that speaking rates of both vowels and consonants in spontaneous speech are significantly faster than those in read speech. In our previous experiment for recognizing spontaneous presentations, it was found that speaking rate was variable even within a sentence and therefore it was effective to model local speaking rate variation using stochastic models such as dynamic Bayesian networks (Shinozaki and Furui, 2003).

Although it is quite obvious that human beings effectively use prosodic features in speech recognition, how to use them in automatic speech recognition is still difficult. This is mainly because prosodic features are difficult to extract automatically and correctly from speech signal and difficult to model due to their dynamic natures. This is especially true for spontaneous speech.

This paper has focused on corpus-based spontaneous speech recognition issues mainly from the viewpoint of human-to-human monologue speech processing (Category II). Most of the issues discussed in this paper, however, are expected to be applicable to another important category, human-computer dialogue interaction (Category III).

Acknowledgments

The author would like to thank all members of the "Spontaneous Speech" project as well as students at Tokyo Institute of Technology and Kyoto University who contributed to building the large-scale corpus and creating the new technologies described in this paper.

References

Akita, Y., Nishida, M. and Kawahara, T., 2003. Automatic transcription of discussions using unsupervised speaker indexing. In: Proc. IEEE Workshop on Spontaneous Speech Processing and Recognition, Tokyo, pp. 79-82.

Christensen, H., Gotoh, Y., Kolluru, B. and Renals, S., 2003. Are extractive text summarization techniques portable to broadcast news," In: Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, St. Thomas, pp. 489-494.

Evermann, G.. et al., 2004. Development of the 2003 CU-HTK conversational telephone speech transcription system. In: Proc. IEEE ICASSP, Montreal, pp. I-249-252.

Furui, S., 2003a. Recent advances in spontaneous speech recognition and understanding. In: Proc. IEEE Workshop on Spontaneous Speech Processing and Recognition, Tokyo, pp. 1-6.

Furui, S., 2003b. Toward spontaneous speech recognition and understanding. In: Pattern Recognition in Speech and Language Processing. Chou, W., Juang, B.-H. (Eds.), CRC Press, New York, pp. 191-227.

Furui, S., 2004. Overview of the 21st century COE program "Framework for Systematization and Application of Large-scale Knowledge Resources". In: Proc. International Symposium on Large-scale Knowledge Resources, Tokyo, pp. 1-8.

Furui, S., Nakamura, M., Ichiba, T. and Iwano, K., 2005. Analysis and recognition of spontaneous speech using Corpus of Spontaneous Japanese. In: Speech Communication (to be published)

Gauvain, J.-L. and Lamel, L., 2003. Large vocabulary speech recognition based on statistical methods. In: Pattern Recognition in Speech and Language Processing. Chou, W., Juang, B.-H.(Eds.), CRC Press, New York, pp. 149-189.

Hori, C. and Furui, S., 2003. A new approach to automatic speech summarization," In: IEEE Trans. Multimedia, pp. 368-378.

Hori, C., Furui, S., Malkin, R., Yu, H. and Waibel, A., 2003. A statistical approach to automatic speech summarization," In: EURASIP Journal on Applied Signal Processing, pp. 128-139.

Hirschberg, J., Bacchiani, M., Hindle, D., Isenhour, P., Rosenberg, A., Stark, L., Stead, L., Whittaker, S. and Zamchick, G., 2001. SCANMail: Browsing and searching speech data by content. In: Proc. Eurospeech 2001, Aalborg, pp. 2377-2380.

Ichiba, T., Iwano, K. and Furui, S., 2004. (in Japanese) Relationships between training data size and recognition accuracy in spontaneous speech recognition. In: Proc. Acoustical Society of Japan Fall Meeting, 2-1-9.

Janin, A., Ang, J., Bhagat, S., Dhillon, R., Edwards, J., Macias-Guarasa, J., Morgan, N., Peskin, B., Shriberg, E., Stolcke, A., Wooters, C. and Wrede, B., 2004. The ICSI meeting project: Resources and research. In: Proc. NIST ICASSP 2004 Meeting Recognition Workshop, Montreal.

Kawahara, T., Kitade, T., Shitaoka, K. and Nanjo, H., 2004. Efficient access to lecture audio archives through spoken language processing. In: Proc. Special Workshop in Maui (SWIM).

Kawahara, T., Nanjo, H. and Furui, S. 2001. Automatic transcription of spontaneous lecture speech, In: Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, Madonna di Campiglio, Italy.

Kikuchi, T., Furui, S. and Hori, C., 2003. Two-stage automatic speech summarization by sentence extraction and compaction. In: Proc. ISCA-IEEE Workshop on Spontaneous Speech Processing and Recognition, Tokyo, TAP10.

Kolluru, B., Christensen, H., Gotoh, Y. and Renals, S., 2003. Exploring the style-technique interaction in extractive summarization of broadcast news," In: Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, St. Thomas, pp. 495-500.

Lussier, L., Whittaker, E. W. D. and Furui, S., 2004. Combinations of language model adaptation methods applied to spontaneous speech. In: Proc. Third Spontaneous Speech Science & Technology Workshop, Tokyo, pp. 73-78.

Maekawa, K., 2003. Corpus of spontaneous Japanese: its design and evaluation. In: Proc. IEEE Workshop on Spontaneous Speech Processing and Recognition, Tokyo, pp. 7-12.

Maekawa, K., Kikuchi, H., Igarashi, Y. and Venditti, J., 2002. X-JtoBI: an extended J_ToBI for spontaneous speech. In: Proc. ICSLP 2002, Denver, pp. 1545-1548.

Maekawa, K., Kikuchi, H. and Tsukahara, W., 2004. Corpus of spontaneous Japanese: design, annotation and XML representation. In: Proc. International Symposium on Large-scale Knowledge Resources, Tokyo, pp. 19-24.

Mani, I. and Maybury, M.T. (Eds.), 1999. Advances in automatic text summarization," MIT Press, Cambridge, MA.

Oard, D. W., 2004. Transforming access to the spoken word. In: Proc. International Symposium on Large-scale Knowledge Resources, Tokyo, pp. 57-59.

Sankar, A., Gadde, V. R. R., Stolcke, A. and Weng, F., 2002. Improved modeling and efficiency for automatic transcription of broadcast news. In: Speech Communication, 37, pp. 133-158.

Schwartz, R. et al., 2004. Speech recognition in multiple languages and domains: the 2003 BBN/LIMSI EARS system. In: Proc. IEEE ICASSP, Montreal, pp. III-753-756.

Shinozaki, T. and Furui, S., 2003. Hidden mode HMM using Bayesian network for modeling speaking rate fluctuation. In: Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, St. Thomas, pp. 417-422.

Shinozaki, T. and Furui, S., 2004. Spontaneous speech recognition using a massively parallel decoder. In: Proc. Interspeech-ICSLP, Jeju, Korea, 3, pp. 1705-1708.

Shinozaki, T., Hori, C. and Furui, S., 2001. Towards automatic transcription of spontaneous presentations. In: Proc. Eurospeech2001, Aalborg, Denmark, pp. 491-494.

Uchimoto, K., Nobata, C., Yamada, A., Sekine, S. and Isahara, H., 2003. Morphological analysis of the Corpus of Spontaneous Japanese. In: Proc. IEEE Workshop on Spontaneous Speech Processing and Recognition, Tokyo, pp. 159-162.

Venditti, J., 1997. Japanese ToBI labeling guidelines. In: OSU Working Papers in Linguistics, 50, pp. 127-162.

Waibel, A. and Rogina, I., 2003. Advances on ISL's lecture and meeting trackers. In: Proc. IEEE Workshop on Spontaneous Speech Processing and Recognition, Tokyo, pp. 127-130.

SADAOKI FURUI is currently a Professor at Tokyo Institute of Technology, Department of Computer Science. He is engaged in a wide range of research on speech analysis, speech recognition, speaker recognition, speech synthesis, and multimodal human-computer interaction and has authored or coauthored over 400 published articles. He is a Fellow of the IEEE, the Acoustical Society of America and the Institute of Electronics, Information and Communication Engineers of Japan (IEICE). He is President of the International Speech Communication Association (ISCA). He has served as President of the Acoustical Society of Japan (ASJ) and the Permanent Council for International Conferences on Spoken Language Processing (PC-ICSLP). He has also served as a Board of Governor of the IEEE Signal Processing Society, and Editor-in-Chief of Speech Communication as well as the Transaction of the IEICE. He has received numerous awards from IEEE, IEICE, ASJ and the Japanese Minister of Science and Technology. E-mail: furui@cs.titech.ac.jp.

CHALLENGES FOR ADAPTIVE CONVERSATIONAL AGENTS

Kristiina Jokinen
University of Helsinki, Finland

Abstract

State-of-the-art dialogue technology has reached a level that allows its use in commercial applications, enabling users to have short conversations with the system to search for information. However, the unique requirements that arise from the combination of system knowledge representation, interaction management, and interface design have to be addressed as part of the system development process. Especially, challenges that concern adaptation, multimodality, cooperation and social interaction of intelligent agents with human users need to be examined further. In this paper I will discuss these challenges, and consider the prospects for future dialogue systems that will offer natural, intuitive and rich conversational possibilities.

Keywords: adaptation, dialogue systems management, conversational agents, human-computer interaction, natural language, user interfaces, mobile systems

1. Introduction

Computers have become such an essential part of our daily environment that it is difficult to avoid dealing with them in one way or another: we readily use cash machines, library catalogues, electronic banking facilities, information providing systems, assisting translation systems, emails, web browsers, etc. Speech interface has also provided a new range of applications for people and situations where typing or mousing is cumbersome or not possible at all: car navigation, telephone services, home appliances, as well as computer services for people with special needs. Other interaction techniques have also been developed so as to allow interaction by pointing and touching and boosting possibilities for enhanced multimodal interaction. Consequently, the notions of adaptivity, agenthood, and cooperation have surfaced as important issues to be addressed to when building interactive systems that take various users into account. Given the more complex environment in which we have to interact with various automatic services, it has become obvious that adaptation and rich interaction capabilities are not required only from human users but from the systems as well.

As I have already argued (Jokinen 2000), some of the main issues in the attempts to build intelligent dialogue systems concern knowledge acquisition and social cooperative interaction: the system should be able to learn through interaction and build internal knowledge of how to reach one's goals. In this paper I will discuss challenges for building spoken dialogue systems which use natural language and aim at flexible interaction capabilities, adaptation and knowledge management. I use the term 'dialogue system' to refer to an interactive system that maintains special models for representing information and uses the representations to process information at different abstract levels of meaningful knowledge. On the other hand, speech interfaces are a part of a software component that allows the user to issue commands to the system by using speech, and also allows the system to give its responses in speech, but it does not produce abstract concepts in order to support meaning analysis of the user utterances.

The starting point for my discussion is the change in the metaphor used to describe the user's interaction with the system. While the computer has been regarded as a tool, the complexity of the applications and tasks that dialogue systems are employed for has contributed to the emergence of a new view of the computer as an agent which is capable of mediating between the user and the complex application (see discussion e.g. in Jokinen, Raike 2003). This view brings with it various challenges which I will be discussing in Section 2.

The view of the computer as an agent also calls attention to the system's adaptation capabilities. Adaptation has been especially considered in hypermedia research where the user can navigate and search for information. For instance, Brusilovsky and Maybury (2002) describe three major adaptation technologies in these applications: (1) adaptive content selection, (2) adaptive navigation support, and (3) adaptive presentation. In spoken dialogue systems, adaptation has mainly concerned dialogue strategies that allow flexible interaction with the user (Litman, Pan 1999; Chu-Carrol, Nickerson 2000). From early on, research has focused on the user's beliefs and knowledge so as to help the system to provide information that would be appropriate to the user's level of expertise (Chin 1989). In Section 3, I will discuss various issues related to adaptation and its desirability.

Finally, communication does not only include verbal but non-verbal aspects as well. Besides speech, the systems should also pay attention to gestures and expressions, i.e. dialogue systems should support multimodal interaction. In Section 4, challenges related to multimodal interaction will be discussed.

2. Dialogue management, learning and knowledge sources

The computer-as-an-agent-metaphor presupposes that the system can maintain conversation and produce intelligent and relevant responses. The system thus needs to understand spoken language, not only simple speech commands but utterances which can be long, fragmental, and contain hesitations, false starts, repetitions and other deviations from the written standards. Moreover, the system must convince the user that the system responses are reliable in their content and communicatively adequate, i.e. the system is a trustworthy partner which provides true information. Attention should thus be paid to suitable presentation techniques: system responses should be clear and simple and also show understanding of different knowledge and skill levels of the users.

All these issues pose considerable challenges for dialogue management. The current technology allows us to build dialogue systems which interact with the users fairly robustly albeit in a fixed way, and different dialogue management techniques can be distinguished (Jokinen 2003; cf. also McTear 2004). The simplest one is scripted dialogue management which defines appropriate actions at each dialogue point with the help of predefined scripts. However, the technique does not distinguish dialogue structure from domain knowledge, and thus each dialogue state and each possible ordering of the dialogue acts must be explicitly defined in the script. Although fairly sophisticated dialogues can be produced, the approach becomes untenable, if more complex dialogues are to be modelled than basic question-answering on limited domains. A more flexible approach is possible by using forms or frames which define information needed to complete the underlying task. The form-based dialogue management is suitable for tasks where actions can be executed in various orders and the dialogue be driven by the information needed. The form also provides a dialogue context in which the actions can be interpreted and planned so as to allow more varied utterances. Finally, conversational dialogue management aims at improving computer interaction by taking into account human conversational capabilities and by building models for their computational treatment. Previous research has focused especially on AI-based modelling of the speaker's intentions, topic tracking, response planning, cooperation, errors and misunderstandings. This approach also calls for advancement in computational techniques, and various machine-learning techniques have been applied to conversational phenomena.

Besides the model that comprises communicative principles and rules for taking turns with the user, a dialogue system also includes models for the system's knowledge of the world and the application. The world model is usually built according to what is important for the application, and the system's reasoning is tailored to simple inferences in the application domain. This of course limits the system's portability to other domains as well as adaptation to different user preferences and knowledge levels. There is thus a need for extending the system's knowledge so that domain ontology and information from real-size databases could be maximally exploited in the interaction management. To do this, it is necessary to equip the system with a wider knowledge about the world, with effective means to learn more, and to reason about the information.

A better understanding of how coordination and cooperation work in conversations is also necessary. The principles of Ideal Cooperation (see e.g. Allwood 1976; Allwood et al. 2000) describe how rationality, the agents' coordinated actions and mutual trust underlie successful communication. The knowledge comprises factual information about the activity and tasks, roles and attitudinal information about the partners as well as rules for reasoning and ethical consideration that guide the agent's decision making. It also presupposes that the communicative competence is embodied in a partner that the agent is interacting with. Transferred into human-computer interaction, this paves way to the research area that deals with animated virtual agents. Although it is highly controversial whether a natural partner can be realised through an animated interaction agent on the screen, the funny little creatures seem real enough to give an interactive system a more natural flavour than interfaces which simply use a screen, mouse and keyboard.

It should be emphasised that the mere use of spoken language implies human-like conversational capabilities and strengthens the computer-as-an-agent metaphor. However, we need to be careful when assigning qualities like "naturalness" to various means and modes through which interaction takes place. For instance, keyboard and mouse are the most natural means to interact with computers, whereas speech/signs are the most natural media for humans communicate with each other. Theoretical considerations of interaction are necessary in order to train models and construct systems that exhibit sophisticated conversational capabilities, but also experiments with the existing systems are needed. The users acquainted with certain existing ways of interaction become "biased" as to what is natural, and it is not conspicuous what kind of interaction techniques will be developed further into more natural ones and how popular will their reception be. In fact, it is not impossible that the "unnatural" ways of today's interaction will be natural tomorrow.

From the view-point of learning interaction strategies, the useful patterns are learnt via experience. Communicative and strategic knowledge does not only result from the agent's engagement in situations where knowledge simply accumulates through positive feedback. Rather, the lack of success of one's communicative action also provides feedback of how to modify one's behaviour and strategic knowledge to manage other similar situations. Negative situations range from slight misunderstandings to total failure of cooperation, and their impact is of course relative to the importance of smooth communication in the given situation and may also depend on the cultural context (e.g. whether clarification questions are encouraged and whether the partners share the same principles for polite and accurate communication). Learning by giving positive and negative feedback can be modelled by reinforcement learning (Barto, Sutton 1993), which has been applied to dialogue management to learn initiative strategies (Litman et al. 2000; Walker 2000; Scheffler, Young 2002), and also explored in building adaptive speech interfaces and user models (Jokinen et al. 2002b).

3. Adaptation

The problem in adaptive interfaces seems to be the notion of adaptivity itself: adaptation involves learning, learning involves interaction, and interaction changes the knowledge through which adaptation takes place. The first question in adaptation is thus when to adapt and what to adapt to. Since the complex nature and limited models of adaptation, the design and implementation of adaptive interfaces in HCI has mainly focussed on clear and transparent interaction where the users, possessing the full capability of human adaptation, can easily adapt themselves to the system and the usage situation. As the users learn how to operate a particular system, they become accustomed to the interface and may find any changes in the interface distracting. The task that the system is meant to assist with is often simple in that it does not require interactive capabilities on the system side – the user is in full control of how to proceed with the task and the computer functions as an assisting tool.

However, a common problem in interactive systems is that the system's knowledge of the user and the task is restricted: this undervalues the user's versatile competence which varies depending on the task at hand. The notion of adaptivity thus becomes important when discussing interactive systems that take various users into account. In practical systems, especially when considering spoken dialogue interfaces to large databases (e.g. Internet or various business, administrative, library, and educational catalogues) that are used by users with varied knowledge and different cultural background, it is important to take into account requirements for complex interaction: the users' knowledge and intentions, variation in viewpoints and interests, the whole context in which the interaction takes place. In these cases it would be beneficial if the system also tries to adapt to the user. The simplest way to realise the system's adaptivity is to allow the users to make static choices: select colours, sounds, interests, and other preferences, and store these characteristics in personal profiles. On-line adaptation can be realised in the system's ability to classify users into various categories, e.g. according to their navigation choices, so as to provide adapted answers to the user queries. Adaptation can also be extended to user actions in communicative dialogue situations: by allowing the system to monitor the user's behaviour, it is possible to model the user's changed knowledge and expertise and to provide responses and reactions tailored in accordance with the observations (Jokinen, Kanto 2004). Thus adaptation would involve learning via interaction and selecting view-points on the basis of recent dialogue events.

It must be noticed that adaptation can also take place with respect to different applications, devices, and the environment, without explicit verbal communication or mechanical control to request the chage of state. The ambient and ”aware-of” technology provides us with intelligent detectors which can be used to control lights, water, temperature, etc. as well as to send small messages confirming their location, movement, and state. The devices can be extended to act as an extra memory for the users in their daily activities, and it is also possible to build systems that attempt to detect the user’s mood and adapt to the user’s emotional state e.g. by switching on relaxing music if the user is angry. On the hardware level, the system might also be able adapt itself to different software programmes and change itself to be a radio, a phone and a computer as necessary for the user.

When considering the system’s adaptation to the user, we must pay attention to several parameters that center around the user. First of all, the users have different habits and preferences and the system may need to find intelligent ways to classify these as well as determine appropriate adaptation strategies. In fact, this is one of the most studied areas in adaptation: various recommendation systems and e-commerce applications exploit the induced knowledge of the users’ preferences, and in collaborative filtering method, the user’s preferences are compared to the preferences of a group of users with similar background, thus enabling mixing of preferences and likings among the group members who have similar background.

Another much studied areas in user-centred parameters is the user’s attitudes and intentions, concerning appropriate communicative strategies and dialogue management habits that they exploit. The user’s may be active and initiative taking or passive and wait for topics to appear. If the system can adapt its dialogue strategies according to the user’s preferred style, it can be expected that the user would find it easier to follow recommendations and suggestions that the system provides. Another aspect related to user attitudes is the user’s temperament and style: the users vary in their frustration levels, and the system may learn to tune its responses to the user’s mood.

In building adaptive system, some technical and architectural parameters are also important. The accuracy of speech and gesture recognizers is crucial in getting the input data correct, the fusion component where the input streams are combined needs to provide reliable information, and the interpretation and decision making should produce results which are efficient and minimize the processing effort, even in cases where ambiguous situations or conflicts arise. Inaccuracies and errors in the output of the separate system components will only cumulate as the interpretation goes higher up in the abstraction level.

Support for adaptivity must also come from the system architectures which should exhibit flexible plug-and-play platform for experimenting with various adaptive constraints and features. Agent-based architectures, e.g. GALAXY-II (Seneff et al. 1999), CMU Communicator (Rudnicky et al. 1999), TRAINS (Baylock et al. 2002) introduce asynchronicity in the functioning of the system components, thus providing flexibility and freedom from the tight pipeline processing. Notice that “agent” refers to an autonomous software agent, a piece of software that realises a subtask of the application that the system has been built for, and is “intelligent” in the sense of being able to judge and also make decisions of what kind of information is needed for the task completion. In the projects Interact (Jokinen et al. 2002a) and DUMAS (Jokinen, Gambäck 2004) we have used the Jaspis architecture (Turunen, Hakulinen 2002).

Further requirements for adaptive dialogue system come from the functionality of the system and the design of system output. As has been already mentioned, the traditional HCI design principles emphasize clarity and unambiguous phrasing in the system responses. Our experiments confirm the requirement for clarity and consistency (Jokinen, Kanto 2004): even for a competent user it may be confusing if the system responses sound illogical or inconsistent. Besides consistency, the system output should pay attention to the information that is being exchanged. Often it is not only the amount of information that is important but also the kind of information provided. For instance, in the studies by Paris (1988), it was noticed that novice users benefited of descriptions of the machines while experienced users wanted to know about the functionality of the machines. Giving information on any topic thus requires decision about how much details and helpful extra information will be given to the partner as well as how this is going to be presented (cf. Jokinen, Wilcock 2003). Summarization and conclusions may be suitable in contexts where the user can look for more information if she is interested in doing so, while the level of abstraction may be crucial if the user is not familiar with particular terms and thus requires further explanation.

Finally, adaptation leads to usability issues where the main question is the desirability of adaptation. While adaptation is an intriguing and interesting research topic from the point of view of dialogue modelling and human communication, there has been a long debate in the system design and HCI communities about the benefits of adaptation: does it increase the system's usability? We must distinguish the two notions of usability and usefulness: the former refers to the system's properties that make it easy to use, while the latter refers to system's usefulness with respect to resolving a particular task and whether the system's existence is necessary or helpful to the user tackling with the task. The common standpoint emphasizes the computer's role as a tool for which a most appropriate, suitable, flexible interface is the main goal. The user should thus be in control of adaptation and decide when the system should adapt or whether it should adapt at all. The computer-as-agent view-point supports adaptation and bases its arguments on the fact that applications become so huge that it is impossible for a user to master the whole complex system and so a helpful adaptation on the system side would be necessary.

Tangled with the desirability issue is the question of how adaptation should take place. Even if adaptation is considered desirable, even necessary, it may still be preferable for the user to be in charge of adaptation instead of having automatic adaptation. In the ideal case, of course, automatic adaptation would support intuitive interfaces which adapt to the various users providing natural interaction without the user even noticing that something special has taken place in the system's behaviour. After all, this is what happens in human-human communication: adaptation to the partner is so automatic that we do not necessarily notice that it has happened. However, this can also be seen as a seed for mistrust. In a similar way as humans can be unreliable and lie, an adaptive dialogue system could also lie. Unless the user is omniscient, it is difficult to know for sure if the information given by the system is truthful and undistorted. On the other hand, as in human-human communication, also in communication with the computer agents, the partners base their interaction on mutual trust. In accordance with the principles of Ideal Cooperation, the rational agents usually provide truthful information since this is the best strategy that pays off for the agent in long term. Thus, when designing intelligent dialogue systems which aim at agent-like conversational capabilities, we must also make sure that the rationality principles that allow humans to act according to Ideal Cooperation, also hold for the computer interface agents. Another question is then how this kind of trust is created and maintained in communication, especially when the partner is an artificial computer agent, but the discussion on these issues goes beyond the scope of this paper.

4. Multimodal Interaction

In recent years, multimodal interactive systems have become more feasible from the technology point of view, and they also seem to provide a reasonable and user-friendly alternative for various interactive applications that require natural human-computer interaction. Although speech is by far the most natural mode of interaction for human users, there is also a lot of factual information (feedback dealing with understanding, acceptance, surprise, etc.) and tacit information (attitudes, mental states, emotions, preferences, background, identity group, etc.) that are conveyed by non-verbal cues such as facial expressions, gestures, and posture. Furthermore, in many cases, like giving instructions of how to get to a particular place, natural way of interaction is to combine verbal communication with gestures and pointing. Also sign language requires visual mode and gesture recognition, and e.g. Jokinen and Raike (2003) point out that multimodal interfaces have obvious benefits for users who cannot use the common speech or keyboard communication modes. Interesting new avenues for multimodal interaction and technological means to implement multimodality were presented in the PhD-course and seminar organized by the Nordic Multimodal Interaction network MUMIN, reported in Jokinen et al. (2003).

In human-computer interaction multimodality refers to the use of different input/output channels through which data is received and transformed to higher-level representations so that manipulation of the environment can take place (Maybury, Wahlster 1998). One of the first multimodal systems was Put-That-There -system (Bolt 1980) which allowed the users to interact with the world through the projection on the wall by using speech and pointing gestures. In the CUBRICON (Shapiro 1988), one could use speech, keyboard, mouse on text, maps, and tables, and the system aim at flexible use of modalities in a highly integrated manner. Many present-day multimodal systems concentrate on speech and graphics or tactile input information. For instance Oviatt et al. (2000) studied the speech and pen system Quickset, and within a cooperation project supported by the Finnish national technology agency TEKES, speech and graphical point-and-click interface are integrated into a PDA-based multimodal route-navigation system. In another paper in this same volume (Hurtig, Jokinen 2005), we discuss the MUMS-system and natural interaction in its design. The SmartKom project (Wahlster et al. 2001) was a technological project which focused on building a large multimodal system that would allow the user to interact with the system in various home and sightseeing situations through an intelligent interface agent.

Multimodal interactive systems are mainly experimental prototypes which require elaboration and sophistication in order to reach the state of robust technology. Jokinen and Raike (2003) discuss various advantages and disadvantages of multimodal interfaces, and conclude that the main benefit seems to be the freedom of choice: e.g. it is easier to point to an object than talk about it, and the users may also have personal preferences and special needs of one modality over another. On the other hand, multimodal interfaces require coordination and combination of modalities which still needs consolidation as robust technology, and from the point of view of usability, there is a danger that the users are exposed to cognitive overload by the stimulation of too many media.

The assumption behind natural interaction is that the users need not spend much time in training themselves on how to use digital devices and a particular application; instead, they could exploit the strategies learnt in normal human-human communication and thus interaction with the system would be as fluent, easy and enjoyable as possible. Extending dialogue management to multimodal dialogue management thus requires understanding of how multimodal conversation takes place among humans. It must be noticed that even though multimodal capability improves system performance, the enhancement seems to apply only on spatial domains, and it remains to be seen what kind of multimodal systems would assist in other, more quantitative domains like giving feedback. Multimodality is an area of growing interest, and more research and experimentation based on a general theory of communication is needed to show how robust interfaces can be built. A natural language interface can thus provide a user-friendly way to communicate with the computer, and combined with other modalities like graphics and speech, a natural language interface is a powerful tool in human-computer communication.

5. Conclusions

In a few years' time we need to interact with a more complex environment which will not consist of only people with varied knowledge, mixed cultural background, and several languages, but also of computers which are embedded in our daily environment, which can sense our presence and act in a meaningful way. The main challenge in the 21st century Communication Technology Society is thus to integrate engineering with language technology research, and to equip electronic devices with natural interaction capabilities. In this paper I discussed some challenges for building dialogue systems that would exhibit intelligent and natural interaction capabilities. While technological development allows us to construct more sophisticated systems, there is simultaneously a need for more intelligent software which takes into account requirements for complex interaction: the users' knowledge and intentions, variation in their viewpoints and interests, the context in which interaction takes place.

References

Allwood, J. 1976. Linguistic Communication as Action and Cooperation. Gothenburg Monographs in Linguistics 2. Göteborg University, Department of Linguistics.

Allwood, J.; Traum, D.; Jokinen, K. 2000. Cooperation, Dialogue, and Ethics. In: *Special Issue of the International Journal for Human-Computer Studies*.

Barto, A.; Sutton, R. 1993. Reinforcement Learning. MIT Press, Cambridge, Massachusetts.

Baylock, N.; Allen, J.; Ferguson, G. 2002. Synchronization in an Asynchronous Agent-based Architecture for Dialogue Systems. In: Jokinen, K.; McRoy, S. (eds.). *Proceedings of the 3rd SIGDial workshop on Discourse and Dialogue*. Philadelphia. 1–10.

Bolt, R.A. 1980. Put-that-there: Voice and gesture at the graphic interface. In: *Computer Graphics*, 14(3). 262–270.

Brusilovsky, P.; Maybury, M.T. 2002. From adaptive hypermedia to the adaptive Web. In: *Communications of the ACM* 45(5). 30–33.

Chu-Carroll, J.; Nickerson, J. S. 2000. Evaluating Automatic Dialogue Strategy Adaptation for a Spoken Dialogue System. In: *Proceedings of NAACL-00*.

Hurtig, T.; Jokinen, K. (2005). On Multimodal Route Navigation in PDAs. (This volume.)

Jokinen, K. 2000. Learning dialogue systems. In: L. Dybkjaer (ed.). *From Spoken Dialogue to Full Natural Interactive Dialogue – Theory, Empirical Analysis and Evaluation*, *LREC,* Athens. 13–17.

Jokinen, K. 2003. Natural Interaction in Spoken Dialogue Systems. In: *Proceedings of the HCI International Conference (Workshop on Ontologies and Multilinguality in User Interfaces*), Crete Greece. Vol 4. 730–734.

Jokinen, K; Gambäck, B. 2004. DUMAS – Adaptation and Robust Information Processing for Mobile Speech Interfaces. In: *Proceedings of the 1st Baltic Conference Human Language Technologies – The Baltic Perspective*, Riga, Latvia. 115–120.

Jokinen, K.; Kanto, K. 2004. User Expertise Modelling and Adaptivity in a Speech-based E-mail System. In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics ACL-04*, Barcelona, Spain.

Jokinen, K.; Kerminen, A.; Kaipainen, M.; Jauhiainen, T.; Wilcock, G.; Turunen, M.; Hakulinen, J.; Kuusisto, J.; Lagus, K. 2002a. Adaptive Dialogue Systems - Interaction with Interact. In: Jokinen, K. and McRoy, S. (eds.). *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*, Philadelphia. 64–73.

Jokinen, K.; Raike, A. 2003. Multimodality – technology, visions and demands for the future. In: *Proceedings of the 1st Nordic Symposium on Multimodal Interfaces*. Copenhagen.

Jokinen, K.; Rissanen, J; Keränen, H; Kanto, K. 2002b. Learning interaction patterns for adaptive user interfaces. In: *Proceedings of the 7th ERCIM UI4All Workshop*, Paris.

Jokinen, K.; Räihä, K-J.; Paggio, P.; Jönsson, A. 2003. Multimodal Interactions – The MUMIN Workshop and PhD course. *NorFA Yearbook 2003*.

Jokinen, K.; Wilcock, G. 2003. Adaptivity and Response Generation in a Spoken Dialogue System. In: van Kuppevelt, J. and Smith, R.W. (eds.). *Current and New Directions in Discourse and Dialogue*. Kluwer Academic Publishers. 213–234.

Litman, D.; Kearns, M.; Singh, S.; and Walker, M. 2000. Automatic Optimization of Dialogue Management. In: *Proceedings of the 18th COLING*. 502–508.

Maybury, M.; Wahlster, W. 1998. *Readings in Intelligent User Interfaces*. Los Altos, California: Morgan Kaufmann.

McTear, M. 2004. Spoken Dialogue Technology: toward the Conversational User Interface. Springer Verlag: London.

Neal, J.G.; Shapiro, S.C. 1991. Intelligent Multi-media Interface Technology. In: J.W. Sullivan and S.W. Tyler (eds.). *Intelligent User Interfaces, Frontier Series*. New York: ACM Press. 11–43.

Oviatt, Sharon; Cohen, P.R.; Wu, L.; Vergo, J.; Duncan, L.; Suhm, B.; Bers, J.; Holzman, T.; Winograd, T.; Landay, J.; Larson, J.; Ferro, D. 2000. Designing the User Interface for Multimodal Speech and Pen-based Gesture Applications: State-of-the-Art Systems and Future Research Directions. In: *Human Computer Interaction* 15(4). 263–322.

Paris, C. 1988. Tailoring Descriptions to a User's Level of Expertise. In: *Journal of Computational Linguistics* 14 (3). 64–78.

Rudnicky, A.; Thayer, E; Constantinides, P.; Tchou, C.; Shern, R.; Lenzo, K.; Xu, W.; Oh, A. 1999. Creating natural dialogs in the Carnegie Mellon Communicator

System. In: *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech-99),* Budapest. 1531–1534.

Scheffler, K.; Young, S. 2002. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In: *Proceedings of Human Language Technology*. 12–18.

Seneff, S; Lau, R.; Polifroni, J. 1999. Organization, communication, and control in the GALAXY-II conversational system. In: *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech-99)*, Budapest. 1271–1274.

Turunen, M; Hakulinen, J. 2002. Jaspis – A Framework for Multilingual Adaptive Speech Applications. In: *Proceedings of the 6th International Conference on Spoken Language Processing*, Beijing.

Wahlster, W.; Reithinger, N.; Blocher, A. 2001. SmartKom: Multimodal Communication with a Life-Like Character. In: *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001).* Aalborg, Denmark.

Walker, M. 2000. An Application of Reinforcement Learning to Dialogue Strategy Selection in a Spoken Dialogue System for Email. In: *Journal of Artificial Intelligence Research* 12(12). 387–416.

KRISTIINA JOKINEN is Professor of Language Technology at the University of Helsinki and a Visiting Fellow of Clare Hall at the University of Cambridge. Her research concerns AI-based spoken dialogue management, intelligent interactive systems, adaptive interfaces, rational agents, and natural cooperative communication. She developed the Constructive Dialogue Model approach for interaction management in dialogue systems and has directed several national and international research projects on spoken dialogue systems and adaptive user modelling. E-mail: kristiina.jokinen@helsinki.fi

HOW TO SURVIVE IN A MULTILINGUAL EU

Steven Krauwer
ELSNET / Utrecht University (The Netherlands)

Abstract

In this paper we describe briefly why the multilingual nature of Europe poses a problem especially for the smaller languages, and we indicate what could and should be done to alleviate the problem.

Keywords: multilinguality, human language technology, EU programmes, resources

Roles of language

Our language is our most important instrument for communication with others. Where in the past the circle of others would normally remain limited to people living in our direct environment (neighbourhood, city, country) the creation and expansion of the EU have made us all member of a much larger community, where more than 20 languages are being used for communication between citizens. Contrary to the situation in the past we all have to face the fact that most of our fellow EU citizens do not speak or understand our language. This affects a number of aspects of our daily and professional life, and we should ask ourselves to what extent this may cause problems or disadvantages for some of us, and – more importantly – how Human Language Technology (abbreviated HLT) could help to overcome the problems.

Politically we see that more and more of our local policies are determined by EU legislation coming from Brussels. Although the decision procedures are democratic and every member state gets its chances to participate in the discussions leading to legislative measures and is allowed to use its own language at all formal sessions one may wonder whether everybody's voice is heard equally well during this process and the preparatory stages, where informal discussions may be held in one of the major working languages. At this moment HLT is used by the EU to support professional translators and interpreters, and to provide quick and dirty translations of internal documents between some languages. In spite of these efforts there is no guarantee that all EU legislators are playing on an equal playing field as far as language is concerned.

Economically we can now observe that Europe has become our home market, and the world at large our foreign market. In order to be able to sell products and services both on our home and our foreign market we will always have to cross language barriers. In many countries users of services expect to be addressed in their own language, and very often national legislation requires user manuals to be provided in the national language.

From a cultural point of view we have now become part of the European culture. From an integration point of view it is desirable that our cultural heritage is accessible to our fellow EU citizens, and that they have access to ours. Unfortunately much of this heritage is based on or described in language, which constitutes a major obstacle for mutual cultural exchanges.

Our society and economy become more and more information driven. Unfortunately most information is encoded in language, which means that having electronic access to information is a necessary but not a not sufficient condition for having full access to our information society.

Individuals from all member states have become European citizens and can now move freely around in Europe, but one can wonder what it means to be a European citizen if one cannot communicate with most fellow EU citizens. Taking away political frontiers is one step, taking away the language barriers is a natural next step.

Where does that leave us?

The ability to cross language barriers is essential for the integration of Europe and for further economic development of the EU as a whole. This is more pressing for small language communities than for the larger ones. If one’s native language is German (90 million speakers), English, French or Italian (60 million speakers each) there is a lot less exposure to foreign languages, especially since 32% of the EU citizens speak English as their first foreign language (and French and German 10% each).

One can easily live in one of those countries without ever realizing that there exist people who speak a different language. All books are translated, movies are dubbed, and if president Putin opens his mouth on television a voice-over will take over from him within half a second.

According to recent EU figures ca 55% of EU citizens do not have enough command of a foreign language to participate in a conversation.[1]

Traditionally we have three methods to help us to cross language barriers: human translators for written language, human interpreters for spoken language, and (last but not least) learning a foreign language. The first two methods are valid and effective in some situations, but not always applicable in day-to-day communication. The third method can be very helpful in certain situations, and EU programmes that have been set up to promote foreign language learning can have a significant impact on the possibility for EU citizens to communicate with each other, but language learning requires a long term investment, and there are limits to the number of languages one can learn in a lifetime.

The role of HLT

In the absence of feasible alternatives employing HLT seems to be the only way to improve the situation. Over the years the EU has invested massively in the development of HLT, and many dedicated HLT programmes have had a significant impact on the advancement of HLT, including applications oriented towards solving the multilinguality problem. Even though the 6th Framework Programme does not have specific HLT oriented action lines, many of the present projects address language issues.

[1] See e.g. http://europa.eu.int/comm/education/policies/lang/languages/index_en.html

Unfortunately the strong industrial bias of recent EU programmes has led to a situation where the major part of the funding for language and speech technology goes to the major languages. This is not surprising, as industrial players will prefer to invest in the development and deployment of technologies for larger markets. As a consequence there has been only marginal support for the development of HLT for the smaller languages. As the cost development of such technologies is independent of the number of speakers of a language ("*all languages are equally difficult*") this has created a very unbalanced situation.

What can we do to improve the situation at the EU political level?

At the EU political level it is important that the speakers of smaller languages don't accept that their languages (and the speakers themselves) be marginalized in Europe. It is well-known that the cost (both in time and in money) of multilinguality for the EU is enormous, and that it will be hard to resist the temptation to reduce the number of official working languages to just a couple. One may be forced to resort to such or similar pragmatic solutions, but representatives of the smaller (or maybe rather commercially not attractive) languages should under all circumstances try to avoid that such pragmatic solutions put them in a disadvantaged position in comparison with those who will be able to use their native languages on all occasions.

It is mandatory to keep the multilinguality problem on the EU agenda as a top priority, and a common responsibility. In this context one should keep in mind that the biggest potential enemy is the so-called subsidiarity principle. There is nothing wrong with the principle as such ("*don't treat anything at the EU level that could be treated at the national level*"), but in past discussions with EU officials this same principle has been used to explain why the EU could not possibly provide financial support for the technological development of smaller languages, as a language is primarily the responsibility of the national government. This attitude does not only do injustice to the fact that multilinguality is primarily a European problem (as opposed to a collection of national problems), but it also does not seem to be completely consistent with the fact that effectively most of the EU funds for HLT are used to support a few major languages.

One would hope that the next EU Framework Programme will recognize the language dimension of Europe, and will address support for HLT development explicitly, irrespective of the economic potential of a language or EU world leadership ambitions.

What can be done at the national level?

Human resources

As speakers of smaller languages we have to face the facts: if we don't take care of our languages no one will do it – or Microsoft (provided they judge the potential market interesting enough to make the investment).

In order to properly develop HLT for one's own language (both from a monolingual and from a multilingual point of view) a number of preparations are necessary. First of all language and speech technology have to find their way to higher education curricula. Traditionally language technologists tend to come from a linguistics background, whereas speech technologists have an engineering background. Very few of them have received an education directly aimed at language or speech technology, and there is very little integration between the two. Researchers in more recently emerging areas (multimodality, interfaces, knowledge engineering) have even to a larger extent been obliged to educate themselves, as no standard curricula exist for these fields. Reflection on future curricula seems desirable, in order to be able to offer the next generation of researchers and developers in these (interdisciplinary) fields a better tailored package of knowledge and skills. In this context we would like to point to the European Masters Programme, which started as an ELSNET initiative, and which aims at defining (and continuously updating) a 1 year masters curriculum in language and speech technology.[2] The idea is that the curriculum definition serves as a template that can be adopted partially or entirely by the participating institutions. Student exchanges are possible (if local conditions permit), and an annual summer school brings together all participants from all sites.

When building up local expertise with respect to the national language it is important to keep in mind that even if every language is unique, many problems may manifest themselves in several (often related) languages, and may have been solved there. Even if these solutions might not be directly applicable to one’s own language, it is often easier to port the solutions than to try to solve the problem from scratch. In order for researchers to optimally benefit from this it is very important that they get the opportunity to attend international conferences, workshops or courses. The organization of local (or regional) training courses is a very useful instrument to introduce new technologies that have been developed elsewhere.

Language resources

Language resources (written and spoken corpora, lexicons, parsers, annotation tools, etc) are essential for the development of language technologies and for the training of students. These resources, whatever their nature, have all in common that they are expensive (in time and money) to create. In order to maximally exploit the resources that have been and will be created their re-usability is a very important feature. Funders of the creation of resources should take great care to ensure that once these resources have been created for a specific purpose (e.g. a project) they can be re-used by future projects. This has different aspects:

(i) from an IPR point of view it should be ensured that resources created through public funding can be re-used by others without any legal constraints, at least for research purposes;

(ii) technically these resources should be created in conformity with existing standards or best practice, in order to ensure optimal interoperability with other tools and resources;

(iii) organisationally it should be ensured that a body is identified that is responsible for the maintenance and further distribution of these resources, in order to guarantee that these precious materials do not get lost when research teams are dissolved or new hard- and software platforms emerge.

Given the emergence of statistical methods in all sub-areas of HLT there is virtually no limit to the amount of resources researchers can use. As the creation of such resources can be a significant financial burden ELSNET, in cooperation with a number of partners, including ELDA (Paris), CST (Copenhagen), CNR-ILC (Pisa), is in the process of developing the BLARK concept. BLARK stands for Basic Language Resource Kit, and it aims at defining the minimal collection of resources that is needed to do any research and (precompetitive) development in HLT at all. In its final form it

[2] See http://www.cstr.ed.ac.uk/euromasters

should comprise a list of necessary components (specified both qualitatively and quantitatively), and the standards (formal or de facto) to be adhered to. We will also aim at including cost estimations for the production of the various component, based on experience. The BLARK concept was first launched in the ELRA Newsletter published in May 1998.[3] The definition allows for adaptations to specific properties of languages.

The BLARK definition should be used as a common reference point for language communities that want to start their own HLT activities, and that need to make up a priority list of what is needed. Once the definition is available teams can make an inventory of what exists and what is missing.

Initial BLARK definitions have been provided for the Dutch language, by researchers associated with the Dutch Language Union. A first inventory and an identification of priorities has led to a large HLT programme funded by the Dutch government and the regional Flemish government in Belgium.

In the framework of the EU funded NEMLAR project[4] a BLARK definition is being prepared for Arabic. A first draft has been published on the same site.[5]

What can be done internationally at the EU level

Many countries have a long and well-established tradition of national HLT programmes. Within the framework of the creation of the ERA the EU aims at better coordination between national HLT related programmes. A recent initiative is the Lang-Net proposal, that has been submitted by a large consortium of representatives of research ministries from various member states and regions. The objective of this proposal is to make the first moves towards an ERA-Net in the field of language and speech technology. The result of the evaluation of the proposal is not known yet, but in general it is recommendable for countries to try to join such international research coordination activities. If the Lang-Net proposal is successful other countries may join this initiative and (more importantly) the ERA-Net proposal that will be prepared during this action.

What sort of HLT solutions are we looking for?

It is easy to say that we should resort to HLT in order to get our multilinguality problems out of the way, but how realistic is this? In spite of all the efforts made by the R&D community machine translation (MT) is still not mature enough to be accepted as a generally applicable solution. For the time being the creation of high quality MT systems is still a wonderful research topic, but nothing more than that.

Yet it has to be kept in mind that even state-of-the-art MT can be useful. The obvious example is just finding out what a mail message or a web page in a foreign language is about. I am receiving hundreds of spam messages per day, but sometimes I am really curious what it is that people are trying to sell me from Russia, Korea or China, and a free on-line MT system is good enough to get an idea.

If you buy an MT system like Systran you can get it almost for free, and the quality is poor, but if you are prepared to spend a bit more it can be customized to your specific needs, and the quality level improves dramatically. Like in the case the cheap inkjet printers and the expensive cartridges Systran's real business is not the MT system but the customization.

[3] Also published on http://www.elsnet.org/blark.html

[4] http://www.nemlar.org

[5] http://www.nemlar.org/Publications/BLARK-final.pdf

If your company has a professional translation department the introduction of an MT system can easily save you 30% on your translation costs. The raw translation is not good enough for publication, but the total process of making the raw translation and having it edited by a professional translator can become a lot cheaper and faster.

For normal citizens MT is not really a useful option to cross language barriers. In order to find good alternatives we have to abandon the idea that one single solution should solve the problem in all situations. Different situations may require different types of solutions, just like in traffic where you can solve the problem that you happen to be in the wrong place by walking, using your bike or car, taking the train or the plane, or just using the phone.

Let me just give a few examples. Many mobile phones or PDAs come with a small camera these days. Why can't I use this to point at the menu in a restaurant in Tallinn, have it OCR-ed, translated and displayed on the screen in my own language? Why isn't my PowerPoint presentation displayed on two screens in parallel, one in English and one in Estonian (by way of – possibly imperfect – subtitles)? Why doesn't the manager of my hotel use a multilingual authoring system to present his announcements in my own language? Why can't I use my mobile phone or PDA to have the spoken word *spinach* translated in Estonian and displayed on the screen so that I can show the shopkeeper that it is spinach I want?

The morale of this should be clear: even if we don't know how to do full MT yet there are lots of ways to deal with the language problem in different contexts, especially since many contexts offer opportunities to support language communication with additional modalities (combination of spoken and written language, gesturing, facial expressions, video displays, etc).

Concluding remarks

I have tried to describe above why multilinguality is a pressing problem, especially for the smaller language communities in Europe. I have also indicated what one could do to in order to keep the problem on the EU's political agenda, what one can do to strengthen one's own local HLT, and what sort of solutions present day HLT can offer.

Personally I do not see an immediate danger that our small languages will disappear in the first hundred years or so, but in my view the real danger is that speakers of smaller languages may find themselves more and more marginalized, both economically and politically, if they don't make a serious effort to overcome the language problem. From my own professional point of view the use of HLT is the most promising direction, but at the same time I would like to make it clear that I also sympathize with the EU's efforts in their language action plan 2004–2006 to encourage people to learn at least two other EU languages in addition to their native language!

STEVEN KRAUWER studied mathematics in Utrecht, and has been researcher and lecturer in computational and mathematical linguistics in the department of general linguistics (now Utrecht Institute of Linguistics UiL OTS) at Utrecht University since 1972. His main research interest is machine translation. He has been actively involved in EU projects since 1980. Since 1995 he is the coordinator of the European Network in Human Language Technologies (created with funding from various EU programmes since 1991).

AN OVERVIEW OF SHALLOW XML-BASED NATURAL LANGUAGE GENERATION

Graham Wilcock
University of Helsinki

Abstract

The paper gives an overview of shallow XML-based natural language generation, including XML pipeline architectures, text planning with XSLT templates, and transformations from text plan trees to text specification trees. The work is based on practical experience in a spoken dialogue system, and examples from this system are presented.

1. Introduction

The paper gives an overview of shallow XML-based natural language generation (NLG) including XML pipeline architectures, text planning with XSLT templates, and transformations from text plan trees to text specification trees, using open-source software. The ideas are based on practical experience in developing an XML-based generation component for a spoken dialogue system (Jokinen et al. 2002). Some examples from this system are given in Section 2, and the basic methods are described further in Section 3, followed by more general discussion in Section 4.

1.1. Pipelines

A pipeline is the most widely-used NLG architecture (Reiter and Dale 2000). In XML-based generation it is easy to use a pipeline as powerful methods for organizing XML pipelines are available. For example, Apache Cocoon (Apache Cocoon Project) is an industrial-strength, scalable XML pipeline processor. The pipeline architecture adopted here follows the textbook by Reiter and Dale (2000) as shown in Figure 1.

The interface between text planning and microplanning is a text plan, a tree whose leaves are domain-specific concept messages. The interface between microplanning and realization is a text specification, another tree whose leaves are linguistic phrase specifications. Both the text plan trees and the text specification trees can be naturally represented in XML, as illustrated in Section 2.

1.2. Templates

The status of template-based generation has been debated by NLG researchers (Becker and Busemann 1999). Generally, templates are considered suitable only for shallow forms of generation, in which the templates contain predefined surface strings. However, if template-based means "making extensive use of a mapping between semantic

- Document Planning (or Text Planning)
  - content determination
  - document structuring
- Microplanning
  - lexicalization
  - referring expression generation
  - aggregation
- Realization
  - linguistic realization
  - structure realization (e.g. HTML)

Figure 1: NLG Pipeline

structures and representations of linguistic surface structures that contain gaps" (van Deemter et al. 1999), then templates can also have a role in deeper forms of generation. In either case, NLG templates can be naturally implemented in XML by means of XSLT templates (Wilcock 2001: 2002).

The approach adopted here is to use templates as a good way to create initial text plan trees that contain gaps to be filled later when the concept messages are turned into phrase specifications in the text specification tree. This is not template-based generation, it is template-based text planning. The text plan is then passed through the various stages of the generation pipeline for further processing. The implementation of this form of template-based text planning using XSLT templates is described in Section 3.1.

1.3. Transformations

The text plan tree is transformed into a text specification tree by the microplanning stages, and the text specification tree is transformed into the required output by the realization stages. This requires tree-to-tree transformation processing. Section 3.2 describes an approach in which the required transformations are performed by a sequence of XSLT stylesheets.

When both the text plan tree and the text specification tree are represented in XML, transformation from one to the other is XML-to-XML transformation. When the required output is XHTML for web pages or Java Speech Markup Language (Sun Microsystems 1999) for speech output, the realization stage also performs XML-to-XML transformation.

2. Examples: Spoken Dialogue Responses

Examples of XML-based response generation within a spoken dialogue system are discussed by Jokinen and Wilcock (2003). Some of the examples are repeated here. The generation component, which performs bilingual generation of responses in Finnish and English for a Helsinki bus timetable enquiry system, has been demonstrated by Wilcock (2003). The responses depend on the dialogue context and can vary from full sentences to short elliptical phrases.

For spoken dialogue response generation, content determination is done by the dialogue manager, and document structuring is greatly simplified because the generated response is typically very short. The starting point for the generation component in a spoken dialogue system is therefore a specification of the utterance content which is determined by the dialogue manager, as described in Section 2.2.

2.1. Generation from NewInfo

In the approach taken here, dialogue response planning starts from the new information focus, known as *NewInfo*. This approach to generation from NewInfo was developed by Jokinen (Jokinen et al. 1998; Jokinen and Wilcock 2003). One of the tasks of the generator is to decide how to present the NewInfo to the user: whether it should be presented by itself or whether it should be *wrapped* in a link to the Topic.

(1) User: *Which bus goes to Malmi?*
System: *Number 74.*

(2) User: *How do I get to Malmi?*
System: *By bus - number 74 goes there.*

In Example 1 NewInfo is the information about the bus number, while in Example 2 NewInfo concerns the means of transportation. In both cases, NewInfo is presented to the user by itself, without linking to the Topic.

(3) *When will the next bus leave for Malmi?*

(a) *2.20pm*

(b) *It will leave at 2.20pm*

(c) *The next bus to Malmi leaves at 2.20pm*

Whether NewInfo should be wrapped or not depends on the changing dialogue context. When the context permits a fluent exchange of contributions, wrapping is avoided and the response is based on NewInfo only, as in Example 3a. When the context requires more clarity and explicitness, NewInfo is wrapped by Topic information as in Example 3b in order to avoid misunderstanding. When the communication channel is working well, wrapping can be reduced, but when there are uncertainties about what was actually said, wrapping must be increased as in Example 3c to provide implicit confirmation. These examples are discussed in more detail by Jokinen and Wilcock (2003).

2.2. Input: an Agenda in XML

The starting point for the generation pipeline is an *agenda*, a set of concepts determined by the dialogue manager. In Example 1 *Number 74*, the bus number information is supplied by the dialogue system's task manager, which consults the timetable database. The dialogue manager puts concepts into the XML agenda, as shown in Figure 2.

```
<agenda id="1">
  <concept info="Topic">
    <type>transportation</type>
    <value>bus</value>
  </concept>
  <concept info="Topic">
    <type>destination</type>
    <value>Malmi</value>
  </concept>
  <concept info="Topic">
    <type>bus</type>
    <value>exists</value>
  </concept>
  <concept info="NewInfo">
    <type>busnumber</type>
    <value>74</value>
  </concept>
</agenda>
```

Figure 2: Agenda for Example 1

The root node of the XML tree is <agenda>, and its children are <concept> nodes, which here represent an unordered set. The dialogue manager labels each concept as NewInfo or Topic, using its knowledge of how the concepts relate to the current dialogue situation. These labels are represented in the XML agenda by attributes. Here, the concept 'busnumber' is labelled as NewInfo, and the other three concepts are labelled as Topic. The representation shown in Figure 2 is simplified for clarity.

2.3. A Text Plan in XML

```
<TextPlan id="1">
  <Message>
    <type>NumMsg</type>
    <concept info="NewInfo">
      <type>busnumber</type>
      <value>74</value>
    </concept>
  </Message>
</TextPlan>
```

Figure 3: Text Plan for Example 1

In text planning, the content determination stage simply extracts the concepts from the agenda. As the dialogue manager has already decided the relevant concepts and put them in the agenda, no other content determination is needed. However, the generator decides whether to generate only the NewInfo, or whether to include a Topic link in addition to NewInfo. In the case of Example 1, only NewInfo is generated.

The discourse structuring stage creates a text plan tree as shown in Figure 3 using a form of template-based text planning described in Section 3.1. In Example 1 there is only one message, which is typical in spoken dialogue responses. In multi-paragraph text generation there would be large numbers of messages. The text plans are XML tree structures containing variable slots, which will be filled in later by the microplanning stages. In the text planning stage, the concepts from the agenda are copied directly into the appropriate slots. In Example 1 there is only one NewInfo concept, so only this concept is copied from Figure 2 to Figure 3. This first example is therefore minimal, with one concept in one message.

2.4. A Text Specification in XML

The processing during microplanning is done by a sequence of XSLT transformations, as described in Section 3.2. The text plan tree is transformed into a text specification tree, as shown in Figure 4.

```
<TextSpec id="1">
  <PhraseSpec>
    <subject cat="NP">
      <head>number</head>
      <attribute>74</attribute>
    </subject>
  </PhraseSpec>
</TextSpec>
```

Figure 4: Text Specification for Example 1

The messages in the text plan tree are replaced by phrase specifications in the text specification tree. In the referring expression stage of microplanning, domain concepts are replaced with linguistic referring expressions. In the text specification in Figure 4 the single concept of Figure 3 has been replaced by a linguistic specification.

2.5. Output: Speech Markup in XML

The realization stage produces output marked up in Java Speech Markup Language (JSML) (Sun Microsystems 1999) as shown in Figure 5.

```
<jsml lang="en">
  <div type="sent"> Number
     <sayas class="number">74</sayas> </div>
</jsml>
```

Figure 5: Speech Markup for Example 1

The <head> words of the text specification (Figure 4) provide the main content of the output. In the speech markup, <div type="sent"> marks sentence boundaries and <sayas> tells the speech synthesizer how to say something - in Figure 5 it shows that 74 should be pronounced “seventy-four” not “seven four”.

2.6. Another Example

In Example 1 *Which bus goes to Malmi?*, only the concept ’busnumber’ is labelled as NewInfo, and the other three concepts are labelled as Topic. This leads to the minimal output *Number 74*. In the case of Example 2 *How do I get to Malmi?*, the dialogue manager specifies the means of transportation as NewInfo and the destination as Topic. This will be generated as *By bus*.

In addition, the dialogue manager provides further new information about the bus number, following a co-operative dialogue strategy. This will be generated as *Number 74 goes there*. In this case a more complex text plan is selected as shown in Figure 6. The text plan has two messages, and a prosody markup is inserted between the two messages. The text plan has four concepts. One concept is copied into the first message and three concepts are copied into the second message.

```
<TextPlan id="2">
  <Message>
    <type>TransportMsg</type>
    <concept info="NewInfo">
      <type>transportation</type>
      <value>bus</value>
    </concept>
  </Message>
  <prosody cat="pause"/>
  <Message>
    <type>NumDestMsg</type>
    <concept info="NewInfo">
      <type>busnumber</type>
      <value>74</value>
    </concept>
    <concept info="NewInfo">
      <type>bus</type>
      <value>exists</value>
    </concept>
    <concept info="Topic">
      <type>destination</type>
      <value>Malmi</value>
    </concept>
  </Message>
</TextPlan>
```

Figure 6: Text Plan for Example 2

The text plan in Figure 6 is transformed by the microplanning stages into the text specification shown in Figure 7. The two messages are made into two phrase specifications, with the prosody element between them. Because the *destination* concept in Figure 6 is marked as Topic, it is pronominalized as *there* by the referring expressions stage. If the same destination concept were marked as NewInfo, it would be realized by the actual text value of the destination placename, in this case *to Malmi*.

```
<TextSpec id="2">
  <PhraseSpec>
    <adverbial cat="PP">
      <head>by</head>
      <object cat="NP">
        <head>bus</head>
      </object>
    </adverbial>
  </PhraseSpec>
  <prosody cat="pause"/>
  <PhraseSpec>
    <head>go</head>
    <features>3sg</features>
    <subject cat="NP">
      <head>number</head>
      <attribute>74</attribute>
    </subject>
    <adverbial cat="PP">
      <head>there</head>
    </adverbial>
  </PhraseSpec>
</TextSpec>
```

Figure 7: Text Specification for Example 2

The text specification in Figure 7 is realized in Example 2 as the response *By bus - number 74 goes there*. This response is marked up in JSML as shown in Figure 8. The two phrase specifications are realized as two utterance divisions. The prosody element is realized as a `<break>` element telling the speech synthesizer that a pause is required before the second part of the response.

```
<jsml lang="en">
  <div type="sent"> By bus </div>
  <break size="large"/>
  <div type="sent"> Number
    <sayas class="number">74</sayas>
    goes there </div>
</jsml>
```

Figure 8: Speech Markup for Example 2

3. XML-based Generation

The starting point is the agenda of concepts specified in XML. The finishing point is the utterance to be passed to the speech synthesizer, specified in a speech mark-up language which is also XML. This is generating XML from XML. Moreover, XML is used for all the internal representations along the stages of the pipeline. One advantage of using XML is that the internal representations can be defined in DTDs or XML Schemas, and they can be checked by standard XML validation techniques.

In some stages a new XML tree is created. For example, in the text planning stage described in Section 3.1 a new text plan tree is created. In other stages information is added to the existing tree, or nodes are replaced with new nodes. For example, in the referring expression stage described in Section 3.2 domain concepts are replaced with linguistic referring expressions, within the existing text specification tree.

Whether creating a new XML tree or adding information to the existing XML tree, different processing options can be used. For example, the DOM document model can be used for explicit manipulation of the tree nodes by Java programs, or XSLT can be used to specify XML transformations. It is simple to set up a sequence of transformations, in which the output of one transformation is the input to the next transformation. This is a natural way to implement the NLG pipeline architecture.

3.1. Template-based Text Planning

In the NewInfo-based model of generation described in Section 2.1, text planning selects those concepts marked as NewInfo as the basis for generation, and decides whether NewInfo will be the only output, or whether it will be preceded by the Topic linking concepts. In a less shallow approach, text planning combines messages to construct a text plan. In a more shallow approach, complete text plans are predefined by means of XSLT named templates, as illustrated in Figure 9.

```
<xsl:template name="TRANSPORT-PLUS-NUMDEST">
<TextPlan>
  <Message>
    <type>TransportMsg</type>
    <xsl:copy-of select="./concept[type='transportation']"/>
  </Message>
  <prosody cat="pause"/>
  <Message>
    <type>NumDestMsg</type>
    <xsl:copy-of select="./concept[type='busnumber']"/>
    <xsl:copy-of select="./concept[type='bus']"/>
    <xsl:copy-of select="./concept[type='destination']"/>
  </Message>
</TextPlan>
</xsl:template>
```

Figure 9: Simplified Text Plan Template

The text plan template creates a new XML tree, with root node `<TextPlan>`. This contains two messages. The messages have variable slots, which will be filled in later by the lexicalization and referring expression stages. In the text planning stage, the concepts from the agenda are copied directly into the appropriate slots by means of `<xsl:copy-of>` statements. The example in Figure 9 shows a simplified text plan for Example 2 *By bus - number 74 goes there*.

Selecting the appropriate text plan template is based on the agenda: which concept types are in the agenda, and whether their information status is Topic or NewInfo. The selection can be implemented by means of nested `<xsl:choose>` statements.

3.2. Transformation-based Microplanning

In the microplanning stages of the pipeline, further information needs to be added to the XML tree, or nodes in the tree need to be replaced by new nodes. This can be done either by explicit tree node manipulation using the DOM document model, or by specifying transformations in XSLT stylesheets. We have implemented prototype generators using both approaches, but suggest that these transformations can be most naturally expressed using XSLT, combining XPath expressions to find the relevant part of the tree and XSLT expressions to specify the transformations.

In the referring expression and lexicalization stages, the domain concepts in the text plan are replaced by noun phrases for referring expressions and by other language-specific lexical items. We illustrate how these stages can be implemented in XML by means of the following simplified examples of XSLT templates. The basic idea is that concepts which are marked as Topic are realized as pronouns, whereas concepts which are marked as NewInfo are realized as full descriptions.

```
<!-- REFERRING EXPRESSIONS: PRONOUNS -->
<xsl:template match="concept[@info='Topic']"
              mode="referring-expression">
  <xsl:choose>
  <xsl:when test="type='busnumber'">
    <xsl:text> it </xsl:text>
  </xsl:when>
  <xsl:when test="type='destination'">
    <xsl:text> there </xsl:text>
  </xsl:when>
  <xsl:when test="type='bustime'">
    <xsl:text> then </xsl:text>
  </xsl:when>
  </xsl:choose>
</xsl:template>
```

Figure 10: Referring Expressions: Pronouns

In Figure 10, a destination concept which is marked as Topic is pronominalized as *there*. By contrast, if the same destination concept were marked as NewInfo, it could be realized as a full description by the template in Figure 11, which generates a prepositional phrase with the preposition *to* followed by the actual text value of the destination placename, obtained by the `<xsl:value-of>` statement.

```
<!-- REFERRING EXPRESSIONS: DESCRIPTIONS -->
<xsl:template match="concept[@info='NewInfo']"
              mode="referring-expression">
  <xsl:choose>
  <xsl:when test="type='busnumber'">
    <xsl:text> number </xsl:text>
    <xsl:value-of select="value/text()"/>
  </xsl:when>
  <xsl:when test="type='destination'">
    <xsl:text> to </xsl:text>
    <xsl:value-of select="value/text()"/>
  </xsl:when>
  <xsl:when test="type='bustime'">
    <xsl:text> at </xsl:text>
    <xsl:value-of select="value/text()"/>
  </xsl:when>
  </xsl:choose>
</xsl:template>
```

Figure 11: Referring Expressions: Descriptions

These examples are simplified to show simple text output. In fact the generator performs further syntactic and morphological realization and produces output in a speech mark-up language, as described in Section 3.3.

3.3. Realization

The final stage of the pipeline for text generation listed in Section 1.1 renders the generated text in a specific output presentation format such as HTML. This kind of transformation to a presentation format is the task XSLT was originally designed for.

In spoken dialogue response generation, the output must be in the format required by the speech synthesizer. The implemented demonstration uses FreeTTS (Sun Microsystems 2002), a free, open-source speech synthesizer implemented entirely in Java. FreeTTS accepts input marked up in JSML, Java Speech Markup Language (Sun Microsystems 1999). As JSML is XML-based it can easily be produced by the XSLT pipelines.

However, XSLT is not suitable for all kinds of processing. The Finnish language has highly complex morphology, and a generator for Finnish must be able to handle complex morphological processing as part of the realization stage. XSLT would be unsuitable for this kind of processing, and in any case existing morphological generation software is available, which we wish to re-use. In such situations, XSLT can be combined with general purpose programming languages by embedding extension functions in XSLT templates. These functions can be written in Java (Apache XML Project).

4. Discussion and Related Work

It is easy to set up a pipeline architecture with XSLT. It is easy to perform template-based generation with XSLT. It is easy to perform tree-to-tree transformations with XSLT. This clearly raises the wider question of whether XSLT is really suitable as a general tool for building complete NLG systems.

This is discussed by Cawsey (Cawsey 2000), who concludes that XSLT transformations can be used for generation when the input is fairly constrained, but that XSLT is not suitable for less constrained input, when we need to turn to general purpose programming languages or NLG tools.

White and Caldwell (White and Caldwell 1999) compare their Java-based EXEMPLARS generator with XSLT and suggest that their system has advantages because it is more object-oriented. However, the problem with object-oriented systems and general purpose programming languages is precisely that they require the participation of skilled programmers. One of the major advantages of XSLT is that it is *not* a general-purpose programming language, but falls rather into the category of scripting languages which are (at least relatively) accessible and useable by non-programmers.

In any case, XSLT can be combined with general purpose programming languages by adding Java extension functions to XSLT templates. This means that even where general purpose programming languages are required for specific purposes, a pipeline of XSLT transformations can still be used as a uniform framework.

Seki (2001) has demonstrated an XML-based generation system for Japanese. His work combines Java and XSLT processing in a pipeline architecture similar to the approach discussed here.

Foster and White (2004) describe the use of XSLT for text planning. They combine XML-based generation with an open-source surface realizer implemented in Java.

References

Apache Cocoon Project. Apache Cocoon. http://cocoon.apache.org/

Apache XML Project. Xalan-Java. http://xml.apache.org/xalan-j/

Becker, Tilman; Busemann, Stephan (eds.) 1999. May I Speak Freely? Between Templates and Free Choice in Natural Language Generation. Proceedings of the KI-99 Workshop. DFKI, Saarbrücken

Cawsey, Alison 2000. Presenting tailored resource descriptions: Will XSLT do the job?. In: *9th International World Wide Web Conference*. http: //www9.org/w9cdrom/

Foster, Mary Ellen; White, Michael 2004. Techniques for text planning with XSLT. In: *RDF/RDFS and OWL in Language Technology: Proceedings of the 4th Workshop on NLP and XML (NLPXML-2004)*, Barcelona. 1–8

Jokinen, Kristiina; Kerminen, Antti; Kaipainen, Mauri; Jauhiainen, Tommi; Wilcock, Graham; Turunen, Markku; Hakulinen, Jaakko; Kuusisto, Jukka; Lagus, Krista 2002. Adaptive dialogue systems - Interaction with Interact. In: *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*, Philadelphia. 64–73

Jokinen, Kristiina; Tanaka, Hideki; Yokoo, Akio 1998. Planning dialogue contributions with new information. In: *Proceedings of the Ninth International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Ontario. 158–167

Jokinen, Kristiina; Wilcock, Graham 2003. Adaptivity and response generation in a spoken dialogue system. In: van Kuppevelt, J.; Smith, R. (eds.), *Current and New Directions in Discourse and Dialogue*, Kluwer Academic Publishers. 213–234

Reiter, Ehud; Dale, Robert 2000. Building Natural Language Generation Systems. Cambridge University Press

Seki, Yohei 2001. XML transformation-based three-stage pipelined natural language generation system. In: *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (NLPRS-2001)*, Tokyo. 767–768

Sun Microsystems 1999. Java Speech Markup Language Specification, version 0.6. http://java.sun.com/products/java-media/speech/

Sun Microsystems 2002. FreeTTS: A speech synthesizer written entirly in the Java programming language. http://freetts.sourceforge.net/

van Deemter, Kees; Krahmer, Emiel; Theune, Mariët 1999. Plan-based vs. template-based NLG: A false opposition?. In: Becker and Busemann (1999). 1–5

White, Michael; Caldwell, Ted 1999. Beyond XSL: Generating XML-Annotated Texts with EXEMPLARS. CoGenTex, Inc.

Wilcock, Graham 2001. Pipelines, templates and transformations: XML for natural language generation. In: *Proceedings of the 1st NLP and XML Workshop*, Tokyo. 1–8

Wilcock, Graham 2002. XML-based Natural Language Generation. In: *Towards the Semantic Web and Web Services: XML Finland 2002 - Slide Presentations*, Helsinki. 40–63

Wilcock, Graham 2003. Integrating Natural Language Generation with XML Web Technology. In: *Proceedings of the Demo Sessions of EACL-2003*, Budapest. 247–250

GRAHAM WILCOCK is Docent of Language Technology at University of Helsinki. He is co-organiser and co-chair of the NLPXML series of international workshops on natural language processing and XML. He has given invited talks and courses on XML-based natural language generation in UK, Japan, Finland and Estonia. He is co-chair of the 10th European Workshop on Natural Language Generation (2005). E-mail: graham.wilcock@helsinki.fi.

THE ROLE OF THE WEB IN MACHINE TRANSLATION

Martin Volk
Stockholm University

Abstract

The computerized translation of documents from one language to another language is one of the central research topics in Computational Linguistics. Automatic translation is also undoubtedly of high commercial interest. The omnipresent Web has opened new directions for the online use of Machine Translation (MT) systems, but it also provides a wealth of new resources which can be exploited for and by MT systems.

This talk focuses on how MT systems can profit from harvesting parallel corpora from the Web, which then serve as input for bilingual terminology extraction and example-based MT. We also discuss how the Web can be used for restricting word senses and for deciding between translation alternatives. We maintain that MT in the future will be tightly intertwined with the Web.

Keywords: Web as Corpus, Machine Translation, Parallel Corpora, Parallel Treebanks

1. Introduction

This paper starts from two simple observations:

1. Human translations are better than machine translations.
2. The computer is good at retrieving human translations (much better than at translating itself).

We call these observations, the Translation Memory Lesson, and we will elaborate on this lesson in section 2. These two observations lead us to the question: Where can the computer get human translations from?

Well, human translators who use Translation Memory systems accumulate databases with parallel texts over time. And this is a valuable resource that many do not want to share with others unless they are paid for it. But human translations are also freely available in the web. There is a wealth of web pages, reports, user manuals, laws and announcements plus their translations accessible over the web. Therefore the computer should be asked to find and use those translations first, before being asked to translate itself. The task of finding good human translations in the web is not trivial and we will report on some approaches in section 3.2.

But let us assume that our computer has found a number of documents related to our current translation task (in the right domain, for the right language pair). How will it be able to exploit the parallel corpus?

First the documents need to be aligned on the document level and then on the sentence level. Now if we are to translate a new document, then ideally many sentences from our new document will have an exact match in the parallel corpus. Even almost exact matches (e.g. sentences with differing names or date expressions) are very valuable and provide good translation examples.

But realistically, for many sentences there will not be an exact match nor a fuzzy match. Therefore we need to break the sentences down into chunks (clauses, phrases) and match those units. In order to provide training and evaluation material for aligned phrases we work on parallel treebanks.

1.1. Parallel treebanks

Treebanks have become valuable resources in natural language processing (NLP) in recent years Abeillé (2003). A treebank is a collection of syntactically annotated sentences in which the annotation has been manually checked so that the treebank can serve as training corpus for natural language parsers, as repository for linguistic research, or as evaluation corpus for NLP systems. The current interest in treebanks is documented in international workshop series like "Linguistically Interpreted Corpora (LINC)" or "Treebanks and Linguistic Theories" (TLT). But also the recent international conferences in Computational Linguistics have seen a wide variety of papers that involved treebanks. Treebanks have become a necessary resource for many research activities in NLP.

On the other hand recent years have seen an increasing interest in parallel corpora (often called bitexts). See for example Melamed (2001) or Borin (2002) for a broad picture of this area. The interest in corpus work by translation science and translators is documented in Olohan (2004). But surprisingly little work has been reported on combining these two areas: parallel treebanks. We define a parallel treebank as a bitext where the sentences of each language are annotated with a syntactic tree, and the sentences are aligned below the clause level, typically on the phrase level.

The field of parallel treebanks is only now evolving into a research field. Cmejrek et al. (2003) have built a treebank for the specific purpose of machine translation, the Czech-English Penn Treebank with tectogrammatical dependency trees.

The Nordic Treebank Network[1] has started an initiative to syntactically annotate the first chapter of "Sophies World"[2] in the Nordic languages. Currently a prototype of this parallel treebank with the first chapter in Swedish, Norwegian, Danish, Estonian and German has been finished.

As part of this project Volk and Samuelsson (2004) have annotated 220 Sofie sentences in German and in Swedish and aligned them on the phrase level. The alignment was based on automatic word alignment provided by Jrg Tiedemann (as described in Tiedemann (2003)). Word alignment information was then automatically projected to phrase alignments and manually corrected.

Our work on parallel treebanks has led us to the question of how a future translation system will profit from a parallel treebank, which in turn raised the more fundamental question of how such a system will look like. We maintain that future translation systems will strongly interact with web-based resources. In this paper we present our vision on

[1] The Nordic Treebank Network is headed by Joakim Nivre. See www.masda.vxu.se/∼nivre/research/nt.html

[2] The Norwegian original is: Jostein Gaarder (1991): Sofies verden: roman om filosofiens historie. Aschehoug.

how such a system will exploit the various information sources in the web and combine them into a cascaded approach.

2. The Translation Memory Lesson

It is both a well-known and disturbing fact that professional translators do not like to work with Machine Translation systems. They much rather use Translation Memory (TrMem) systems.

A Translation Memory system by itself cannot perform any translation. It is rather a database which stores sentences pairwise from source and target language. That is, a TrMem system is initially empty. It is filled with sentences and their translated target-language counterparts by a human translator. When a new text is to be translated, the TrMem system checks for each sentence of the new text whether it is already stored in its database. If it finds an exact or an approximate match, the system retrieves the translation, i.e. a previous human translation. In case multiple translations are available, the system will offer the translations of the best matches. Well-known examples of TrMem systems are Trados Translator's Workbench, Star TRANSIT, and Atril's Déjà Vu.

A machine translation system, on the contrary, analyzes every sentence before it synthezises a translation. That means that a given sentence is segmented into its words, the words are reduced to their base forms, and these are searched in a bilingual computer lexicon for grammatical information and for their target language equivalents. Depending on the translation model (direct translation, transfer-based translation or interlingua translation) some intermediary representation is computed.

For example, in the transfer-based approach, the grammatical and functional structure of the source sentence is determined, and it is transferred into the corresponding target language sentence structure, the corresponding words are inserted and the new sentence is generated. If a semantic representation is used between source sentence and target sentence, then this representation called interlingua.

It is obvious that the linguistic analysis required from a machine translation system is much more error-prone than retrieving a sentence from a Translation Memory. But on the other hand, this analysis is also much more flexible. If grammar and lexicon in a machine translation system have a broad coverage, it is possible to translate tens of thousands of different sentences with such a system, whereas a TrMem system will only find the sentences already stored in its database.

Recent years have seen a growing success of TrMem systems. The reason for this success is to be found in the fact that these systems do what a computer does best: remember a vast amount of data, and retrieve them efficiently. Professional translators prefer TrMem over MT systems since they can rely on the TrMem output (human translations), whereas they will have to post-edit many sentences that have been translated by MT. Since the translation engine of most MT systems can only be marginally parametrized or modified by the user, the translator may end up correcting the same mistakes over and over again.

2.1. Translation Memories versus Machine Translation

Machine translation suffers from the many ambiguities in natural language that can only be resolved using semantic features, context or world knowledge. But these knowledge entities are difficult and labor-intensive to come by. Therefore, commercial MT systems contain only the most prominent semantic features and little to none context and world knowledge. Due to this lack they provide only limited translation quality.

TrMem systems are obviously most useful when a source text contains many sentences that have previously been translated. This is typically the case in letters following business transactions (billing, complaints etc.) that remain the same except for some product names, amounts, price and date specifications. Other examples are manuals of updated software that can reuse all translations except for the sections on the updated functionality.

Moreover, TrMem systems are a lot easier to build than MT systems. One needs to implement a powerful database to store the sentence pairs, the matching algorithm (extending the search to similar sentences is the most difficult part), and a pleasant user interface. An additional sentence alignment tool for entering already translated texts helps to increase the usefulness. With these modules one can easily use the TrMem system for numerous different languages. The only limit is given by the support for their respective character sets (umlauts, diacritics, etc.).

On the contrary, extending an MT system to a new language is a very complex task. The vocabulary of this language must be collected and stored systematically in a machine-readable lexicon. Since the minimal size for a useful lexicon is on the order of several 10,000 entries, this can hardly be done from scratch. The wealth of information from printed dictionaries must be exploited. But still, the morphological processes (inflection, derivation, compounding) need to be implemented. Then, the grammar rules of the language must be formalized and special parsers are required. Semantic information needs to be added for nouns, verbs and adjectives in order to reduce the ambiguity in analysis and synthesis. Recently statistical MT has approached and in some cases even surpassed the quality of rule-based MT. But it requires parallel corpora from the respective domain as training material.

Considering all this, we understand that MT systems struggle to find their place between TrMem (sentence storage) and online dictionaries (in particular terminology databases; word storage). MT systems can quickly produce raw translations for information skimming. But in order to improve the translation quality the user has to invest a lot of effort for lexicon updates as well as text preparation (e.g. controlled language input) and post-editing. MT works best if the source text is from a well defined subject area, all words are known to the system, and the sentences are simple (few embedding levels and clear clause boundaries). In other words, an MT system works best if it is clearly tuned to a certain subject area and text type. But if one has to tune the system so intensely, one might be better off to use a TrMem, where one can store complete sentences from a given subject area with their correct translations.

A TrMem is restricted to complete sentences. Only minor modifications can be applied to stored translations, such as date substitutions. MT on the other hand is too flexible. It does not account for the interdependencies of words and constituents. We will sketch a middle pathway between these extremes in this paper.

3. Using the Web

Recently the Web has been used for various tasks in Natural Language Processing. Among them such prominent tasks as word sense disambiguation (see Turney (2004)) or the resolution of ambiguities in attaching prepositional phrases. Concerning the latter, we have reported about our experiments on using web search engine frequencies for solving the prepositional phrase attachment problem for German (see Volk (2001)).

Translators (as many casual users) have found the web a most useful resource for looking up how a certain word or phrase is used. Since queries to standard search engines allow for restrictions to a particular language, it has become easy to obtain usage information which was buried in books and papers (or local databases at best) prior to the advent of the web. We will summarize two other examples of how a translator may profit from the web:

3.1. Translating Compound Nouns

Grefenstette (1999) has shown that WWW frequencies can be used to find the correct translation of German compounds if the possible translations of their parts are known. He extracted German compounds from a machine-readable German-English dictionary. Every compound had to be decomposable into two German words found in the dictionary and its English translation had to consist of two words. Based on the compound segments more than one translation was possible. For example, the German noun *Aktienkurs* (share price) can be segmented into *Aktie* (share, stock) and *Kurs* (course, price, rate) both of which have multiple possible translations. By generating all possible translations (share course, share price, share rate, stock course, ...) and submitting them to search engine queries, Grefenstette obtained WWW frequencies for all possible translations. He tested the hypothesis that the most frequent translation is the correct one. He extracted 724 German compounds according to the above criteria and found that his method predicted the correct translation for 631 of these compounds (87%). This is an impressive result given the simplicity of the method.

3.2. Parallel Texts in the Web

Translation Memory systems have become important for translators. But a TrMem system is of no help when the sentence to be translated is not found in its database. But often previous translations in the respective domain exist and are published in the web. The task is to find these text pairs, judge their translation quality, download and align them, and store them into a Translation Memory.

Resnik (1999) therefore developed a method to automatically find parallel texts in the web. In a first step he used a query to the AltaVista search engine by asking for parent pages containing the string ”English” within a fixed distance of ”German” in anchor text. This generated many good pairs of pages such as those reading ”*Click here for English version*” and ”*Click here for German version*”, but of course also many bad pairs. Therefore he added a filtering step that compares the structural properties of the candidate documents. He exploited the fact that web pages in parallel translations are very similarly structured in terms of HTML mark-up and length of text. A statistical language identification system determines whether the found documents are in the suspected language. 179 automatically found pairs were subjected to human judgement. Resnik reports that 92% of the pairs considered as good by his system were also judged good by two human experts. In a second experiment he increased the recall by not only looking for parent pages but also for sibling pages, i.e. pages with a link to their translated counterpart. For English-French he thus obtained more than 16,000 pairs.

This approach was further elaborated in Resnik and Smith (2003), where they also searched URLs for language names or abbreviations. And they added supervised learning and other measures for determining translation equivalence.

Another approach to finding parallel texts in the web could be to use an MT system. Given a source language document from the web we can use MT to get a rough translation of the document. This rough translation is then used as a query to a search engine for finding human-translated documents in the web. To my knowledge this approach has not yet been investigated.

As more parallel corpora are gathered from the web, the issue of translation quality becomes critical. If those corpora shall be used as input for translation memories, only good translations should be selected. Automatically determining the quality of a given document pair will become an urgent challenge for natural language processing research.

4. The translation system of the future

Many resources for translation help are now available over the web. This includes bilingual and multi-lingual dictionaries, terminology collections and parallel documents. Future translation systems will have to exploit these resources. MT (in the traditional sense) will only be one part of such a translation systems. A next generation translation system will also consist of online dictionary access systems, terminology extraction systems and parallel corpora crawlers.

Given a document to be translated, a future translation system will use it to find similar documents on the web (as can be done with current search engines). It will then check whether translations of those documents can also be found.

In this way parallel corpora will be harvested from the web. They will serve as input to Translation Memory systems, but will also be exploited for example-based machine translation, sometimes also called data-oriented translation (see Carl and Way (2003)). Towards this goal they will be aligned on the sentence and phrase level. Parallel treebanks will help to to train phrase level aligners. Example-based translation will use the aligned phrases (sometimes even complete clauses) for automatic translation.[3]

Such a translation system might generate multiple translation alternatives. And it will use statistical language models to rank the alternatives according to their naturalness with respect to the target language (as is done in statistical MT today).

If the translation system does not gather the appropriate material to translate complete sentences, it will help on the word level with terminology research and access to on-line dictionaries.

5. Conclusions

We have argued for an increased interaction of translation help systems with the web. We believe that the automatic translation system of the future will work with a cascaded approach. First it will look for exact matches and fuzzy matches of complete sentences. If it fails to find such a match, it will back-off to smaller units, i.e. phrases and clauses. Such units are used in example-based translation systems. We are convinced that translation quality increases if the longest possible units are used in the translation process. Deep linguistic analysis with word level segmentation and target sentence generation shall only be used if all other levels failed to provide an appropriate translation.

[3] Some Translation Memory providers are now offering such features. Here a quote from www.atril.com: ”Déjà Vu X’s unique assemble feature combines example-based machine translation ideas with proprietary research. It takes advantage of the fact that, while sentence repetition may be scarce in many types of texts, repetition of smaller fragments is much more common.”

References

Abeillé, Anne (ed.) 2003. Building and Using Parsed Corpora: Vol. 20 of *Text, Speech and Language Technology*. Dordrecht: Kluwer

Borin, Lars (ed.) 2002. Parallel Corpora, Parallel Worlds. Selected Papers from a Symposium on Parallel and Comparable Corpora at Uppsala University, Sweden, 22-23 April, 1999.: Vol. 43 of *Language and Computers*. Amsterdam: Rodopi

Carl, M.; Way, A. (eds.) 2003. Recent Advances in Example-Based Machine Translation: Vol. 21 of *Text, Speech and Language Technology*. Springer

Cmejrek, Martin; Curin, Jan; Havelka, Jiri 2003. Treebanks in machine translation. In: *Proc. Of the 2nd Workshop on Treebanks and Linguistic Theories*, Växjö. 209–212

Grefenstette, Gregory 1999. The World Wide Web as a resource for example-based machine translation tasks. In: *Proc. of Aslib Conference on Translating and the Computer 21*, London

Melamed, I. Dan 2001. Empirical Methods for Exploiting Parallel Texts. Cambridge, MA: MIT Press

Olohan, Maeve 2004. Introducing Corpora in Translation Studies. Routledge

Resnik, Philip 1999. Mining the web for bilingual text. In: *Proc. of 37th Meeting of the ACL*, Maryland

Resnik, Philip; Smith, Noah A. 2003. The web as a parallel corpus. In: *Computational Linguistics* **29(3)**, 349–380

Tiedemann, Jörg 2003. Acta universitatis upsaliensis: Uppsala University

Turney, Peter 2004. Word sense disambiguation by web mining for word co-occurrence probabilities. In: *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3).*, Barcelona, Spain

Volk, Martin 2001. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In: *Proc. of Corpus Linguistics 2001*, Lancaster

Volk, Martin; Samuelsson, Yvonne 2004. Bootstrapping parallel treebanks. In: *Proc. of Workshop on Linguistically Interpreted Corpora (LINC) at COLING*, Geneva

P rofessor Martin Volk is head of the Computational Linguistics group at Stockholm University. He holds a M.Sc. in Artificial Intelligence from the University of Georgia (USA) and a Phd in Computer Science from the University of Koblenz (Germany). His research interest in recent years focused on basic research on corpus-based PP attachment resolution and parallel treebanks, and on applied research in cross-lingual information retrieval, information extraction, and ontologies. As a visiting lecturer, he has recently taught treebanking at the 2004 summer school ”Empirical Methods in Natural Language Processing” at the University of Tartu. E-mail: volk@ling.su.se.

II Session Papers

PHONOLOGICAL AND MORPHOLOGICAL MODELING IN LARGE VOCABULARY CONTINUOUS ESTONIAN SPEECH RECOGNITION SYSTEM

Tanel Alumäe
Tallinn University of Technology (Estonia)

Abstract

This paper describes progress in the development of a large vocabulary continuous speech recognition system for Estonian. The Estonian SpeechDat-like phonetic database, containing telephone recordings from 1332 different speakers, is used for training acoustic models and testing speech recognition performance. Clustered triphones with multiple Gaussian mixture components are used for acoustic modeling. Pseudo-morphemes are used as basic units in a statistical trigram language model. Automatically derived morpheme classes are used to improve the robustness and coverage of the language model. The pronunciation dictionary is generated using context-sensitive rewrite rules. Experimental results show a significant improvement in word error rate with regard to the baseline system.

Keywords: LVCSR, language modeling, pronunciation modeling

1. Introduction

The objective of our work is to develop methods and technologies that allow building practical large vocabulary continuous speech recognition systems for Estonian. Estonian is an agglutinative language, thus it's words are heavily inflected depending on their syntactic role. This makes the number of distinctive words in the language very large. To reduce OOV rate, inflected words can split into stems and endings using a morphoanalytical tool. It has been shown that this approach significantly improves the performance of Estonian LVCSR (Alumäe 2004).

This paper describes our experiments in the domain of large vocabulary continuous speech recognition using the recently completed SpeechDat-like phonetic database for training and testing. Pseudo-morphemes (stems, endings and compound particles) are used as basic units in a statistical trigram language model. The units are automatically clustered into classes in order to increase language model robustness and coverage. To better model the Estonian phonology, a set of context sensitive rewrite rules were developed for generating the pronunciation dictionary. The performance of the LVCSR system is evaluated on the sentence utterances in the SpeechDat-like database.

2. Speech database and text corpora

2.1. The speech database

The SpeechDat-like speech database project was launched at the Institute of Cybernetics in 2002 (Meister et al. 2002). It was aimed to collect telephone speech from a large number of speakers for speech and speaker recognition purposes. More than 1000 speakers were expected to participate in the recordings, later the goal was extended to 2000 speakers. Due to well established design principles, the SpeechDat databases, especially Finnish SpeechDat, was chosen as the prototype for the project.

The main technical characteristics of the database are as follows: sampling rate 8 kHz, 8-bit mono A-law encoding, calls from fixed and cellular phones as the signal source, calls from both home and office environments.

Each recording session consists of a fixed set of utterance types, such as isolated and connected digits, natural numbers, money amounts, spelled words, time phrases, date phrases, yes/no answers, person and company names, application words and phrases, phonetically rich words and sentences.

All acceptable quality recording sessions have been orthographically transcribed. Different noises in the utterances were also annotated. The five different noise types are: filled pauses and hesitations ([fil]); speaker noises, such as lip smacks, laugh, throat clear ([spk]); intermittent noises, such as door slams, music, phone ringing ([int]); stationary noises, such as sound of a car engine or street noise ([stat]); unintelligible speech (**).

The collecting of the database ended in the beginning of 2004. The final number of "good" recording sessions, including truncated but otherwise acceptable sessions, is 2969. As many speakers were asked to call 10 times, the total number of different speakers is 1332. The number of acceptable utterances is 177 793. This totals in about 241.1 hours of audio data. The number of different words in the database is 11 731.

For speech recognition experiments, the database was divided into training and test set. Test set was composed by randomly selecting 300 speakers out of those who only called once.

2.2. Text corpora

For language model training, a part of the Tartu University corpus of Estonian literary language was used. The used part contains approximately 15 million words. Most of the texts come from two national newspapers, "Postimees" and "Eesti Ekspress", and only about 5% from original prose written in 1990s.

For language model evaluation, the transcriptions of the long sentences in the SpeechDat-like speech database were used. The texts are relatively neutral in style, resembling more fiction than newspaper articles.

3. LVCSR system

3.1. Language modeling

To cope with the huge number of different word forms resulting from the agglutinativeness of the Estonian language, pseudo-morphemes are used as basic units in the statistical trigram language model. The language model training texts are processed by the morpheme analyzer (Kaalep and Vaino 2000), which marks the boundaries of word compounds in compound words, and marks the boundaries between stem and ending in inflected word forms. The word forms are split at word compound separators as well as stem-ending separators, if the resulting ending is more than one grapheme long. The endings are tagged so that the they are modeled separately from the stems that have the same orthography. The tagging also enables to reconstruct words forms from shorter segments after decoding. Compound words are modeled as separate words. The approach is similar to the one used in earlier experiments (Alumäe 2004).

The pseudo-morpheme based trigram LM was computed using the SRILM toolkit (Stolcke 2002) using the 60 000 most frequent units as vocabulary. The cutoff value for bigrams and trigrams was 2. Kneser-Ney smoothing method (Kneser and Ney 1993) was used. Discounted n-gram probabilities were interpolated with lower-order estimates. The resulting model has estimates for 1 182 039 bigrams and 1 344 910 trigrams.

In order to better generalize to unseen and rare word sequences, a class-based morpheme trigram model was created. We used the perplexity minimization algorithm (Brown et al. 1992) implemented in SRILM to automatically derive morpheme classes from the training texts and trained a morpheme class trigram model. The number of classes was fixed to 800 based on our earlier experience (Alumäe 2004).

Language model performance was measured on the transcriptions of the sentences in the speech database test set. The corresponding utterances were later used for measuring recognition performance. The perplexities using different interpolation weights are shown on Figure 1(a). The best perplexity (578) was observed when the interpolation coefficient of the class-based model was 0.4. This is a 12% improvement over the pure morpheme model (perplexity 653). The out-of-vocabulary rate of the model is 4.2%.

3.2. Phonological modeling

In earlier systems, we have used a phone set and pronunciation dictionary that is composed almost directly from orthography. This is without doubt not optimal for speech recognition.

The proposed phonological modeling is based on research by Eek and Meister (1999). The basic units are 25 segmental phonemes (9 vowel and 16 consonant phonemes). Long monophthongs and diphthongs are considered as sequences of two segmental phonemes. The only exception here are plosives, which in the proposed system are modeled by different units for short and long duration (this is contrary to Eek and Meister (1999)). The palatalized and non-palatalized versions of many consonants are modeled by the same unit. This is due to the fact that the it is difficult to determine palatalization from word orthography, and the palatalized and unpalatalized versions of a word map to the same orthographic word, making such differentiation obsolete for speech recognition. The spectral differences in palatalized and unpalatalized phonemes are hoped to be modeled well using Gaussian mixtures in acoustic models.

It is well known that Estonian has three distinctive duration degrees. However, listening tests have shown the existence of only two (short vs. long), not three phonological degrees on segmental and syllabic levels. The threefold distinction can be made only if information from the next syllable is available. Thus, the proposed system models only short and long durations, as the overlong duration cannot me modeled with traditional HMM-based approach.

The pronunciation dictionary that implements the phonology was generated using a set of context sensitive rewrite rules. The rules were developed in a test-driven way: pronunciations of about 1000 phonetically balanced words were collected from the documentation of the BABEL Estonian speech database (Eek and Meister 1999). The words were processed through a morphological analyzer that marked the separator between word compounds. Next, new ad-hoc regular expression based rules were added to the rule set until the rule-generated pronunciations of the sample words matched their actual pronunciation. It turned out that most of the rules deal with rewriting plosive durations in various contexts, as this is where the orthographic transcription differs from phonetic transcription the most. The rule set can handle correctly only non-compound words and words were the compound separator has been specially annotated.

3.3. Acoustic modeling

The open source SphinxTrain toolkit was used for training the acoustic models. Models are created for 25 phonemes, the five filler/noise types and silence.

All audio was converted from 8-bit A-law to 16-bit linear encoding, as the feature extractor program cannot handle A-law data. For acoustic features, MFCC coefficients were used. The coefficients were calculated from a frequency band of 130 Hz - 3400 kHz, using a preemphasis coefficient of 0.9. The window size was 0.0256 seconds and the frame rate was 100 frames/second. A 512-point FFT was used to calculate 31 filter banks, out of which 13 cepstral coefficients were calculated.

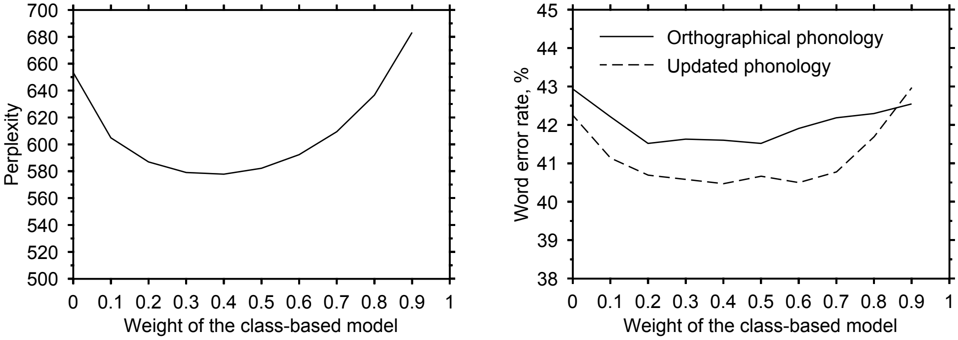
All units are modeled by continuous left-to-right HMMs with three emitting states and no skip transitions. The output vectors are 39-dimensional and are composed of 13 cepstral coefficients, 13 delta cepstra and 13 double delta cepstra.

The training process started with creating of context-independent models which were initialized from flat start. The flat-start models were re-estimated through Baum-Welch algorithm until the improvement in the total log likelihood of the models of the last iteration was less than a given constant. Next, context-dependant untied models were created for all triphones that are present in the training data. Here, word beginning, word ending, word internal and single word triphones are all treated separately. The number of unique triphones was 10380. The untied triphones were cloned from their corresponding monophone models and reestimated until convergence. Once the untied models were computed, decision trees for HMM state tying were built. The linguistic questions for decision trees were created automatically from context independent model statistics using the bottom-up top-down clustering algorithm implemented in SphinxTrain. After the creating of the decision trees, they were pruned to have exactly 8000 unique leaves, which is the number of senones the tied models would have in total. Then, tied-state triphone models were trained. Tied-state models were initialized from corresponding untied models and reestimated. Finally, the models were successively split to have 2, 4 and 8 Gaussian mixture components, and reestimated after each split until convergence.

4. Recognition experiments

The recognition experiments were performed using the Sphinx4 (Lamere et al. 2003) toolkit. For the experiments that used a class-based language model, custom Java classes were developed.

Utterences from the speech database test set were used for recognition. There were totally 2359 sentence utterances in the test set, but in order to reduce the time needed for experiments, only every 7th was used, which resulted in 342 utterances. Error rates including substitution, deletion and error rates were measured for non-compound word units. The language model uses pseudo-morphemes as basic units, but the recognized morphemes were merged back to words before comparing with reference transcriptions.



(a) (b)

Figure 1: Language model perplexity (a) and word error rates (b) using different interpolation weights of the class-based language model.

As the concept of compound words is not implemented in the language model, the reference transcriptions were processed by a morphological analyzer and the compound words were split into separate words.

First, the performance of the baseline system was measured. The baseline system uses a pseudo-morpheme based trigram language model with a simple orthography-based pronunciation dictionary. The word error rate for this system was 42.9%. Then, experiments using the class-based language model interpolated with the morpheme-based model were performed, using varying interpolation weights. The best word error rate was observed when the interpolation weight was 0.5 — the word error rate of 41.5% is a 3.3% relative improvement over the baseline system.

Next, experiments using the proposed phonological modeling were made. The system with a pseudo-morpheme based language model achieved word error rate of 42.2% which is a 1.6% relative improvement over the baseline system. Using the class-based model, the best result was observed at interpolation coefficient of 0.4 (i.e. biased towards the morpheme-based model), which had a word error rate of 40.4% and a word accuracy of 63.9%. This is a 5.8% relative improvement over the baseline system and a 2.7% relative improvement over the similar class-based system that used simple orthography-based phonology.

The word error rates of all experiments are shown on Figure 1(b).

5. Conclusions

We investigated the performance of Estonian large vocabulary speech recognition based on the SpeechDat-like speech database. To reduce the high OOV rate caused by the agglutinativeness of the language, we used pseudo-morpheme based units for language modeling. In order to increase language model robustness and coverage, we automatically induced 800 morpheme classes from text corpora statistics, and interpolated the morpheme-based language model with the class-based model. This resulted in 12% relative improvement in language model perplexity and 3.3% improvement in word error rate.

To better model Estonian word pronunciations, we slightly changed the baseline phone set and used a set of context sensitive rewrite rules for generating the pronunciation dictionary. This together with a class-based language model resulted in our best word error rate of 40.4% which is a 5.8% relative improvement over the baseline results. The recognition accuracy was 63.9%.

The word error rate of 40.4% is quite high for large vocabulary speech recognition. However, it must be noted that the training and test data was telephone speech and often not very intelligible even for human listener. The biggest impact on the high word error rate should lie in language modeling — the perplexity of 578 and out-of-vocabulary rate of 4.2% is high even compared to other similar languages. The main reasons for this should lie in the relatively small size of the training text corpora and the relative complexity and domain-difference of the test sentences.

References

Alumäe, Tanel 2004. Large vocabulary continuous speech recognition for Estonian using morpheme classes. In: *Proceedings of ICSLP 2004 - Interspeech*, Jeju, Korea. 389–392

Brown, Peter F.; Pietra, Vincent J. Della; deSouza, Peter V.; Lai, Jennifer C.; Mercer, Robert L. 1992. Class-based n-gram models of natural language. In: *Computational Linguistics* **18(4)**, 467–479

Eek, A.; Meister, E. 1999. Estonian speech in the BABEL multi-language database: Phonetic-phonological problems revealed in the text corpus. In: *Proceedings of LP'98. Vol II.*. 529–546

Kaalep, Heiki-Jaan; Vaino, Tarmo 2000. Teksti täielik morfoloogiline analüüs lingvisti töövahendite komplektis. In: *Arvutuslingvistilt inimesele*, Tartu. 87–99

Kneser, R.; Ney, H. 1993. Improved clustering techniques for class-based statistical language modelling. In: *Proceedings of the European Conference on Speech Communication and Technology*. 973–976

Lamere, Paul; Kwok, Philip; Walker, William; Gouvea, Evandro; Singh, Rita; Raj, Bhiksha; Wolf, Peter 2003. Design of the CMU Sphinx-4 decoder. In: *Proceedings of the 8th European Conf. on Speech Communication and Technology*, Geneva, Switzerland

Meister, Einar; Lasn, Jürgen; Meister, Lya 2002. Estonian SpeechDat: a project in progress. In: *Fonetiikan Päivät 2002 — The Phonetics Symposium 2002*. 21–26

Stolcke, Andreas 2002. SRILM – an extensible language modeling toolkit. In: *Proceedings of Intl. Conf. on Spoken Language Processing*: Vol. 2, Denver. 901–904

TANEL ALUMÄE was awarded the Master of Technical Sciences Degree at Tallinn University of Technology in 2002. At present he is a researcher in the Laboratory of Phonetics and Speech Technology of the Institute of Cybernetics at Tallinn University of Technology. Scientific interests include: speech and language processing, signal processing, artificial intelligence. E-mail: tanel.alumae@phon.ioc.ee

TOWARDS A PHONOLOGICAL MODEL OF ESTONIAN INTONATION

Eva Liina Asu
Institute of the Estonian Language, Tallinn, Estonia

Abstract

The first systematic study of standard Estonian intonation is presented in Asu (2004) where both read and spontaneous speech are analysed within the autosegmental-metrical framework of intonational analysis. The present paper discusses options of representing the phonology of Estonian intonation by a model capturing various co-occurrence restrictions which characterise the Estonian intonational phrase. First, the possibility of representing Estonian intonational grammar by a linear finite-state model in the form of a transition network such as Pierrehumbert's (1980) model of American English is examined. Second, the structure of Estonian tunes is considered within a hierarchical model such as Ladd's (1996) modification of Pierrehumbert's finite-state grammar. It emerges that a certain differentiation between prenuclear and nuclear accents is needed because of the occurrence patterns of the three types of rising tunes. Also, the representation of Estonian tunes in a model where the prenuclear pitch accents are constrained to be the same is problematic. This is due to certain co-occurrence restrictions particularly in the patterns including low accents (H+L*). A multi-layered finite-state grammar is proposed for the phonological generation of Estonian intonational tunes which could serve as a starting point for further work in developing a probabilistic computational model of Estonian intonation.

Keywords: intonational phonology, autosegmental-metrical theory, modelling, finite-state grammar, prenuclear and nuclear accents, intonational tunes, co-occurrence restrictions, low accentuation

1. Introduction

Since the late 1970s, it has been widely accepted that intonation, like the segmental part of the sound system of languages, is phonologically structured (Ladd 1996). This means that the mapping between the meaning of utterances and the fundamental frequency contour (F0) involves abstract pitch elements, equivalent to the vowels and consonants of segmental phonetics, that mediate between the meaning of the utterance and the actual sound in language-specific ways. The most commonly used phonological framework for intonational studies at the moment is the autosegmental-metrical (AM) theory that represents pitch contours as sequences of discrete intonational events, distinguishing between pitch accents and edge tones. In AM, pitch accents are analysed as consisting of High (H) and Low (L) level tones, or pitch targets, which are associated with metrically strong (stressed) syllables or boundaries. In the original view (Pierrehumbert 1980) all pitch accents consist of a single H or L tone, or a combination of two tones. The tone of a pitch accent which associates to a stressed syllable is indicated with an asterisk, as either H* or L*. In addition to this central (or 'starred') tone, a pitch accent may contain a 'leading' (e.g. H+L*) or 'trailing' tone (e.g. H*+L).

The first systematic description of Estonian intonation within the AM framework is presented in Asu (2004). Based on this analysis the present paper discusses options of representing the phonology of Estonian intonation within a finite-state grammar - a simple generative device that works through an utterance from left to right. In the generation process, after the selection of the first element, the possibilities of occurrence of all other elements are determined by the nature of the elements preceding them (Crystal 1991: 137).

2. Finite-state grammars for English

In Pierrehumbert's (1980) original proposal for American English, tonal sequences for an intonational phrase are generated by a finite-state grammar which is represented as a transition network as shown in Figure 1. According to the grammar, tunes consisting of any combination of pitch accents and any phrase accent and boundary tone are well formed. No co-occurrence restrictions among pitch accents are expressed even if these are in principle not ruled out.

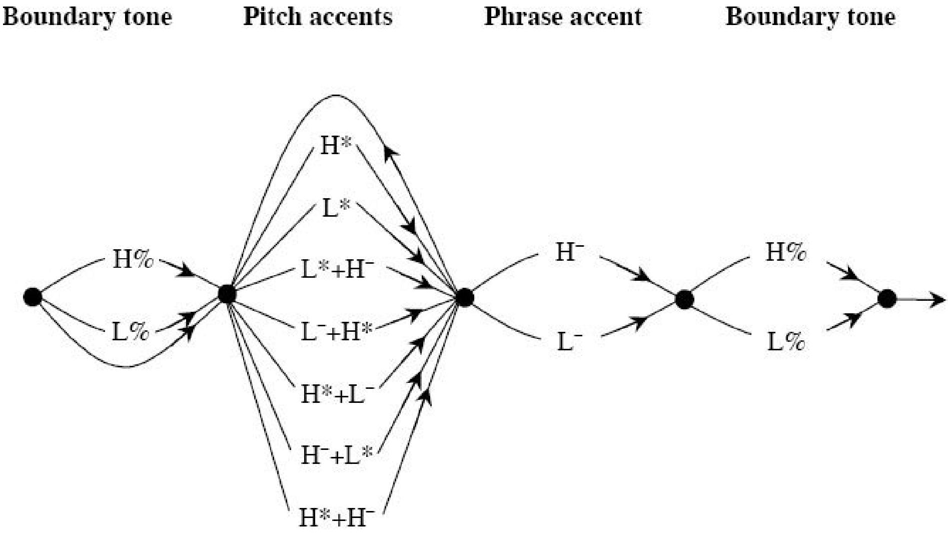


Figure 1. The finite-state grammar of English intonation after Pierrehumbert (1980: 13)

In Pierrehumbert's (1980) model, no distinction is made between prenuclear and nuclear accents. The nuclear accent is merely the last pitch accent in the intonational phrase. After an initial boundary tone any of the seven pitch accents can be chosen and then, as indicated by the leftward pointing arrow on the empty loop, the full inventory is again available. This recursion can be repeated an indefinite number of times, at least in theory.

Ladd (1996: 211), however, argues for the special status of nucleus, which according to him is not incompatible with the basic AM assumption that intonation contours consist of strings of pitch accents. He offers a modification of Pierrehumbert's finite-state grammar that distinguishes between prenuclear accents and the nucleus. Ladd's modified grammar is replicated in Figure 2. According to this grammar a tune consists of at least one accent, the nucleus. This minimal tune is indicated by the rightward pointing arrow on the empty loop under the prenuclear accents. The nuclear accent may be preceded by one or several accents but the grammar allows only for identical prenuclear accents (as indicated by the backward pointing arrows on the small recursion loops). Any nuclear accent can follow any prenuclear accent.

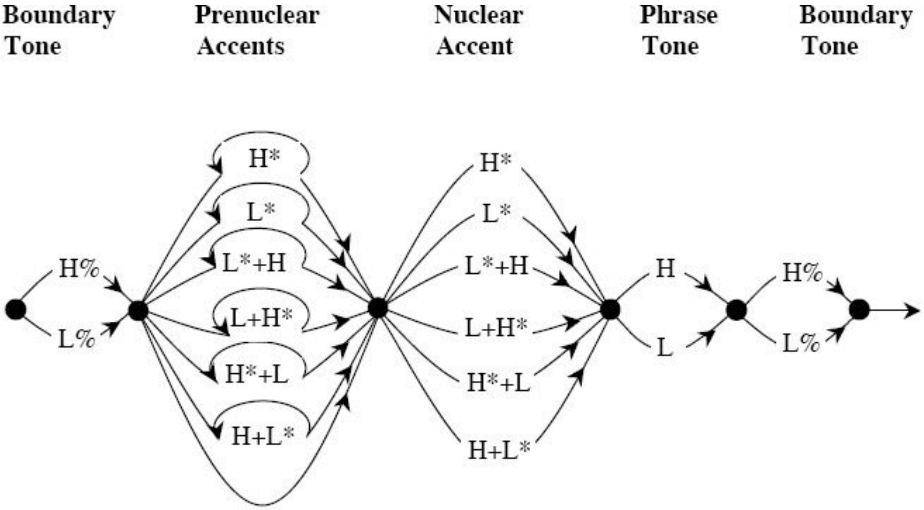


Figure 2. Ladd's (1996: 211) modification of Pierrehumbert's (1980) finite-state grammar

Ultimately, Ladd (1996) is arguing in favour of a hierarchical structure of tunes, which would mean a tune being more abstract than just a string of tones. In doing that he in principle returns to the British tradition of intonational representation (see e.g. Cruttenden 1997) claiming that tunes are structured in terms of nuclear, prenuclear and postnuclear elements (i.e. Nucleus, Head (and Prehead) and Tail in terms of the British tradition). In keeping with the British tradition, Ladd (1996: 218) assigns the nucleus the focus signalling role as the most prominent accent.

Dainora (2001, 2002) proposes a hierarchical model of intonation for American English which unlike Pierrehumbert's (1980) finite-state grammar, but similar to Ladd's (1996) proposal, distinguishes between non-final and final elements in the intonational phrase. Dainora (2001) modifies Pierrehumbert's approach by taking into account the statistics of pitch accent co-occurrence building a probabilistic second order Markov model, which shows that tones have a tendency to combine in predictable patterns.

3. Preliminaries to a grammar of Estonian tunes

The inventory of Estonian pitch accents and boundary tones found in Asu (2004) consists of both monotonal and bitonal accents. In the case of bitonal pitch accents both trailing and leading tones are allowed (i.e. the inventory is mixed-headed). Generally, there is no need for a boundary tone specification in Estonian utterances. The most frequently occurring pitch accent in both nuclear and prenuclear position is an H*+L where an accented high syllable is immediately followed by a fall to a low tone. A sequence of such accents forms the 'default' pattern of Estonian intonation. Other tunes distinguished are the so-called 'stepping-pattern' (similar to that common in English), various patterns containing low accentuation, and rising tunes.

It would be possible to generate Estonian tunes using a grammar similar to that suggested by Pierrehumbert (1980), but in a way which fails to reflect the constraints of the language, because her finite-state grammar generates any possible sequence of pitch accents, occurring and non-occurring alike. On the other hand, Ladd's (1996) finite-state model, presented in Figure 2, severely constrains the structure of the prenuclear part of the tune. In the following, it will be considered how these restrictions are borne out by Estonian tunes.

3.1. Representation of rising tunes and the 'stepping pattern'

An Estonian tune that can be straightforwardly represented in Ladd's (1996) model, allowing only for identical prenuclear accents, is the so-called 'default' pattern consisting of a sequence of H*+L accents. The only obligatory part in this tune is the nucleus.

Similarly straightforward cases for the model to generate are the tunes with different rising nuclear patterns. Asu (2004) distinguished three different rising tunes which were analysed according to the level at which the rise starts (either L* or H*) and the boundary (plateau, i.e. non-specified %, or a rising movement near the boundary, i.e. H%): L*+H %, L* H% and H* %. All three postulated 'rising' nuclear accents can be preceded by one or more H*+L accents or no prenuclear accent. In Asu (2004), the L*+H accent was never shown to appear in the prenuclear position.

The so-called 'stepping pattern' is a tune consisting of a sequence of downstepped H+!H* accents, the first pitch accent of the phrase being an H* rather than an H+!H*. In order to fit this into the model the H* accent has to be underlyingly accounted for as an H+!H*. The nucleus in the 'stepping pattern' after a sequence of (H+!)H* accents can be either an H+!H* or H+L*, which is not problematic for the model because the nucleus can be chosen independently of the prenuclear accents. Allowing for the abstraction of the first prenuclear accent it is possible to account for this pattern in Ladd's model.

3.2. Representation of tunes with low accentuation

Asu (2004) distinguishes between three different tunes containing low accentuation which are labelled depending on where in the utterance the first low accent occurs. The extreme case of low accentuation is the so-called 'low a2/a1' pattern where all the accents are low i.e. the pattern consists of a sequence of H+L* accents. The problem for the representation here is posed by the first accent in the sequence. In the phonological analysis of such patterns in Asu (2004) it was presumed that the H in the first accent is truncated. It is only with this abstraction of allowing the first L* to be 'underlyingly' H+L* that it is possible to fit the 'low a2/a1' tune in the model.

In the so-called 'low a4' pattern the prenuclear accents are H*+L until just before the low nucleus where the last accent before the nucleus is an H*. The representation of these patterns within Ladd's model is possible only when the L in the last H*+L accent is deleted following the tonal linking rules proposed by Gussenhoven (1984), and accordingly the last prenuclear accent surfaces as an H*.

The real problem for Ladd's model is, however, posed by the so-called 'low a3' pattern where there is no unity to the prenuclear component because the pattern changes halfway through the tune: H*+L H* H+L* H+L* H+L*. Even if we allow for the H* to be underlyingly H*+L with a deletion as in the representation of the ‘low a3’ pattern we still have to account for the last two prenuclear accents being different from the first two accents. The ‘low a3’ pattern seems to suggest that the choice for the nucleus is predetermined on accent 3. If the nucleus is H+L* the penultimate accent can be H+L*, and only if the penultimate accent is H+L* can the antepenultimate pitch accent be low, and so on.

Thus, there are two major problems with representing Estonian tunes within Ladd’s (1996) finite-state grammar. First, the recursive loops in the model allow only for a sequence of identical prenuclear pitch accents. Second, the model does not provide for the determination of the nucleus by the prenuclear pitch accent.

4. Finite-state grammar of Estonian intonation

Figure 3 presents an alternative finite-state grammar for generating all the Estonian tunes discussed above. On the whole there does not seem to be evidence that the distinction between prenuclear and nuclear accents plays an important role in Estonian intonational choices. Still, a certain differentiation between the two is needed because the three types of rising tunes do not occur in prenuclear position but as the only or the last accents of the intonational phrase. Therefore, accents that can function both as prenuclear and nuclear are indicated by a recursive loop as these are the ones that the grammar allows to be repeated, whereas those not marked with a recursive loop are the ones that only occur as nuclear accents.

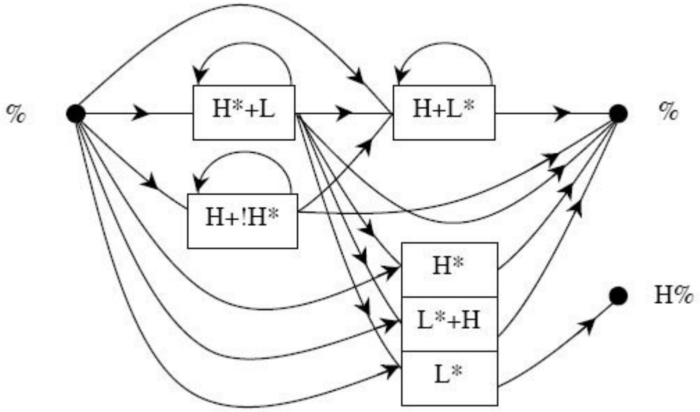


Figure 3. A finite-state grammar to generate Estonian tunes

In the generation of the ‘default’ pattern, the recursive loop on the H*+L accent indicates that the accent can be repeated several times, after which the empty loop connects the accent with the boundary tone. In order to generate the ‘low a4’ pattern the H+L* accent is chosen after one or several H*+L accents as the last accent of the intonational phrase before the boundary. In the ‘low a3’ pattern, the H+L* accent is chosen as accent 3 after two cycles of the H*+L loop, and repeated as many times as needed. In the generation of the ‘low a2/a1’ pattern, the empty loop connecting the initial boundary to the H+L* accent makes it possible to choose the H+L* accent as the first accent of the intonational phrase, and the recursive loop on the accent shows that it can be repeated several times.

5. Conclusion

This paper has discussed the structure of Estonian tunes with a view to representing Estonian intonational phonology within a finite-state model such as Ladd's (1996). It was shown that the representation of Estonian tunes in a model where the prenuclear pitch accents are constrained to be the same is problematic. This is due to certain co-occurrence restrictions particularly in the patterns including low accents. A multi-layered finite-state grammar was proposed for the generation of Estonian intonational tunes listed in Asu (2004). This grammar could serve as a starting point for further work in developing a probabilistic computational model of Estonian intonation.

References

Asu, Eva Liina 2004. The phonetics and phonology of Estonian intonation. University of Cambridge: unpublished doctoral dissertation.

Cruttenden, Alan 1997. Intonation (2nd edition). Cambridge: CUP.

Crystal, David 1991. A dictionary of linguistics and phonetics (3rd edition). Cambridge, MA: Blackwell.

Dainora, Audra 2001. An empirically based probabilistic model of intonation in American English. University of Chicago: doctoral dissertation.

Dainora, Audra 2002. Does intonational meaning come from tones or tunes? Evidence against a compositional approach. In: Bel, B.; Marlien, I. (eds.) *Proceedings of the Speech Prosody 2002 conference, 11-13 April 2002*. Aix-en-Provence: Laboratoire Parole et Langage. 235-238.

Gussenhoven, Carlos 1984. On the grammar and semantics of sentence accents. Dordrecht: Foris.

Ladd, D. Robert 1996. Intonational phonology. Cambridge: CUP.

Pierrehumbert, Janet 1980. The phonetics and phonology of English intonation. Cambridge, MA: MIT.

EVA LIINA ASU has since March 2005 held a post-doctoral fellowship at the Institute of the Estonian Language in Tallinn, having previously worked as a Research Associate at the University of Tartu. She recently received her Ph.D. in Linguistics from the University of Cambridge, UK, where her doctoral dissertation dealt with the phonetics and phonology of Estonian intonation. In addition to phonetics her research interests include language variation and second language acquisition. She is a member of the British Association of Academic Phoneticians and the International Phonetics Association. E-mail: asu@ut.ee

THE “KIEL CORPUS OF READ SPEECH” AS A RESOURCE FOR PROSODY PREDICTION IN SPEECH SYNTHESIS

Caren Brinckmann
Institute of Phonetics, Saarland University (Saarbrücken, Germany)

Abstract

The naturalness of synthetic speech depends strongly on the prediction of appropriate prosody. For the present study the original annotation of the German speech database “Kiel Corpus of Read Speech” was extended automatically with syntactic features, word frequency, and syllable boundaries. Several classification and regression trees for predicting symbolic prosody features, postlexical phonological processes, duration, and F0 were trained on this database. The perceptual evaluation showed that the overall perceptual quality of the German text-to-speech system MARY can be significantly improved by training all models that contribute to prosody prediction on the same database. Furthermore, it showed that the error introduced by symbolic prosody prediction perceptually equals the error produced by a direct method that does not exploit any symbolic prosody features.

Keywords: text-to-speech, database, CART, symbolic prosody prediction, postlexical processes, duration prediction, F0 prediction, perceptual evaluation, German

1. Introduction

The first text-to-speech (TTS) systems relied mostly on rules that were hand-crafted by human experts. For more than a decade, these hand-crafted rules have been successively replaced by models that are automatically trained on annotated corpora with machine learning (ML) methods. Nowadays, intelligibility is no longer a problem for most TTS systems, whereas naturalness can still be improved. One important factor for natural sounding synthetic speech is the prediction of appropriate prosody, i.e. speech rhythm and melody. In many TTS systems the following modules contribute to the prosodic structure of the generated output:

- *symbolic prosody prediction*: prosodic boundaries, accents (location and type of accent), and intonation contours or boundary tones
- *prediction of postlexical phonological processes*: phonemic deletions, replacements, and insertions (e.g. schwa deletion and assimilation of nasals in German), influencing the rhythmic structure of the synthesised utterance
- *prediction of acoustic parameters*[1]: duration of realised phonemes and pauses, F0 (fundamental frequency) of voiced phonemes.

[1] Other acoustic parameters such as intensity or spectral quality could also be predicted.

Many studies concerning ML-based prosody prediction focussed on the improvement of models for one particular prediction task, e.g. symbolic prosody prediction, duration prediction, or prediction of F0 values (cf. Fackrell et al. 1999). Furthermore, the evaluation of the automatically trained models was mostly corpus-based, comparing the predictions of the respective model with the actual realisations in a speech database. However, the implementation of a specific prediction model into an existing TTS system might not result in *perceptually* improved synthetic speech. For example, Brinckmann and Trouvain (2003) compared ML-based duration prediction (a regression tree) with a rule-based duration prediction model (Klatt rules adapted to German). In terms of "objective" corpus-based evaluation measures (RMSE and correlation), the automatically trained regression tree outperformed the Klatt rules. As long as the input to the duration models was optimal, the regression tree was also perceptually superior to the Klatt rules, but when the models were implemented into the German TTS system MARY (Schröder and Trouvain 2003), the perceptual differences disappeared. The main reasons for this masking effect are the inheritance of error in a complex modular TTS system and the fact that not all models contributing to prosody prediction are based on the same data.

The present study uses the German speech database "Kiel Corpus of Read Speech" (KCoRS) comprehensively for all prosody prediction tasks. The KCoRS comprises over four hours of labelled read speech and is available on CD-ROM (IPDS 1994). As described in Section 2, the original annotation of the KCoRS was extended with additional features that were added mainly with pre-existing tools. On this extended database, several classification and regression trees (CARTs) were automatically trained for all prosody prediction tasks. The corpus-based evaluation measures are given in Section 3. The perceptual evaluation described in Section 4 showed that the output of MARY can be significantly improved by training all models that contribute to prosody prediction on the same database. Furthermore, it showed that the error introduced by symbolic prosody prediction perceptually equals the amount of error produced by a direct method that does not exploit any symbolic prosody features.

2. Database

The textual material of the KCoRS consists mostly of isolated sentences taken from a variety of contexts: train timetable queries, phonetically balanced material, and two very short stories. In total, these are 624 sentences, containing 4932 word tokens and 1673 word types. The recordings of two speakers (male speaker *kko/k61* and female speaker *rtd/k62*), who read the entire material, were used for this study.

The segmental labelling of the KCoRS is essentially phonemic with some phonetic additions (e.g. plosive release phase, glottalisation, and nasalisation). Deviations of the realised form from the lexical phonemes (i.e. deletions, replacements, and insertions) are annotated. Orthography, punctuation marks, as well as sentence and word boundaries are also included in the annotation.

The prosodic annotation incorporates the following domains: lexical stress, accent, intonation contour, prosodic phrase boundaries, and pauses. The accent labels include information about accent location and type (6 categories), degree of accentuation (4 categories), and upstep. The phrase-final intonation contours are labelled with 5 different main categories.

The original annotation was automatically extended with the following features:

- sentence type: statements, exclamations, and 6 different question types
- part-of-speech tags, assigned with the statistical tagger TnT (Brants 2000)
- syntactic phrases of limited depth, assigned with a statistical chunk tagger (Skut and Brants 1998) and the SCHUG parser (Declerck 2002)
- grammatical functions of syntactic phrases, assigned with the SCHUG parser
- word frequency measures from the lexical database CELEX (Baayen et al. 1995)
- syllable boundaries, assigned with a simple algorithm based on standard phonotactic principles of German.

3. Prosody prediction with CARTs

CARTs[2] were trained – using the data of speaker *kko/k61* and *rtd/k62* separately – for the prediction tasks listed in Table 1. For the prediction of postlexical processes and acoustic parameters two types of trees were trained: The first type (*Symbolic*) uses symbolic prosody features, whereas the second type (*Direct*) predicts all segmental features without preceding symbolic prosody prediction (see Figure 1).

Mean evaluation measures (averaged over both speakers' trees) are given in Table 1. Accuracy of accent type prediction is rather low (56.2%), but closer inspection revealed e.g. that the three peak categories are mostly confounded with one another. The acoustic parameters were predicted as z-scores, so the correlation coefficients are also given in terms of z-scores. For a detailed description of input features, training regime, and results, see Brinckmann (2004).

Table 1: Mean evaluation measures (across both speakers) for the trained CARTs

prediction task	*Symbolic*	*Direct*
symbolic prosody:	accuracy	
prosodic boundary	95.4%	–
degree of accentuation	88.5%	–
accent location	92.9%	–
accent type	56.2%	–
phrase-final contour	77.8%	–
postlexical processes:	accuracy	
type of change	93.7%	93.1%
replacement rule	93.2%	93.4%
acoustic parameters:	correlation	
duration	0.59	0.56
median F0	0.73	0.64
last F0 in phrase	0.72	0.53

4. Perceptual evaluation

The corpus-based evaluation measures implicitly assume the realisations of one particular speaker as gold standard. However, usually there are several acceptable ways to produce an utterance, and listeners may have differing idiosyncratic preferences. In order to avoid implementing "improvements" to the TTS system that are not accepted by the listeners, the predictions were evaluated by measuring subjective listener preferences with the Comparison Category Rating (CCR) method of ITU-T recommendation P.800 (ITU-T 1996).

One female and one male diphone-based MBROLA voice (Dutoit et al. 1996) implemented in MARY were used to synthesise 20 sentences. These 20 sentences were randomly selected from the KCoRS and had not been used as training, validation or test items for the CARTs.

[2] All CARTs can be downloaded from http://www.brinckmann.de/KaRS/

Three different prosody prediction methods were used to synthesise each sentence:

- MARY: original MARY system (using hand-crafted rules for prosody prediction)
- *Symbolic*: phoneme identity, duration and F0 values are predicted with CARTs, including intermediate symbolic prosody prediction
- *Direct*: direct prediction of phoneme identity, duration and F0 values with CARTs, *without* using any symbolic prosody features.

In addition, every sentence was copy-synthesised by taking the values for phoneme identity, duration and F0 directly from the respective realisation in the KCoRS.

The TTS architecture used for *Symbolic* and *Direct* is shown in Figure 1. The architecture of the original MARY system is almost identical to *Symbolic* except for some minor differences in the syntactic analysis and the fact that the original MARY system implements all prosody prediction modules with hand-crafted rules instead of CARTs.

The synthesised stimuli were presented to 30 native German speakers by pairs A-B or B-A, where A was copy-synthesised and B used one of the three different prosody prediction methods. The listeners had to judge the overall quality of the second sample relative to the overall quality of the first sample using a 7-point scale (from 3 = *much better* to -3 = *much worse*). The comparison opinion scores (COS), which are presented in terms of the order A-B, were used to compute comparison mean opinion scores (CMOS) for each prosody prediction method, synthesis voice, and listener group.

An analysis of the results (with ANOVA and Tukey HSD) showed that MARY (CMOS -1.55) received significantly lower ratings than both *Symbolic* (-0.76) and *Direct* (-0.80). As shown in Figure 2, only 15.4% of all MARY stimuli have a COS of 0 or higher, whereas 38.9% *Direct* and 39.4% *Symbolic* stimuli are rated having a similar or better quality than the copy-synthesised utterance from the KCoRS. *Symbolic* and *Direct* did not differ significantly for either MBROLA voice.

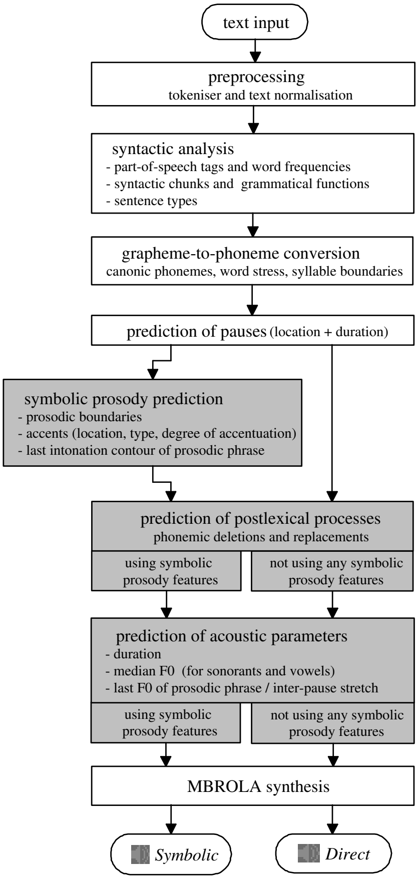

text input
preprocessing
tokeniser and text normalisation
syntactic analysis
- part-of-speech tags and word frequencies
- syntactic chunks and grammatical functions
- sentence types
grapheme-to-phoneme conversion
canonic phonemes, word stress, syllable boundaries
prediction of pauses (location + duration)
symbolic prosody prediction
- prosodic boundaries
- accents (location, type, degree of accentuation)
- last intonation contour of prosodic phrase
prediction of postlexical processes
phonemic deletions and replacements
using symbolic prosody features
not using any symbolic prosody features
prediction of acoustic parameters
- duration
- median F0 (for sonorants and vowels)
- last F0 of prosodic phrase / inter-pause stretch
using symbolic prosody features
not using any symbolic prosody features
MBROLA synthesis
Symbolic
Direct

Figure 1: TTS architecture with *Symbolic* and *Direct* prosody prediction. The shaded modules are implemented with CARTs

Listeners having no or little prior experience with speech synthesis generally gave higher ratings (CMOS -0.98) than regular users or synthesis experts (-1.11). CMOS for MARY was especially low for experts/regular users (-1.71).

5. Discussion and outlook

The perceptual evaluation showed that all three prosody prediction methods mostly received negative COS. Thus, one can generally take the copy-synthesised natural utterances as gold standard. However, there are some exceptions where copy-synthesis was rated as inferior to a prediction method. This could be due to idiosyncratic preferences of the listeners or to the limitations of MBROLA synthesis. For example, the two German MBROLA voices do not allow separate modelling of plosive closure and release, even though plosive releases are deleted much more often in German speech than the respective closures. Furthermore, intensity and spectral quality of the concatenated diphones cannot be controlled.

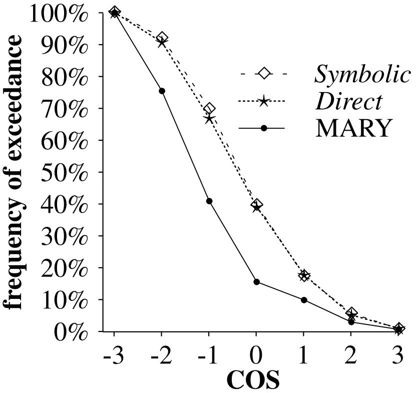


Figure 2: Relative frequency of exceedance of comparison opinion scores (COS), i.e. percentage of ratings that are greater than or equal to the respective COS value, for the three prosody prediction methods

Both ML-based prosody prediction methods *Symbolic* and *Direct* were found to be perceptually superior to the original rule-based MARY method. This shows that the output of a TTS system can be significantly improved by training *all* models that contribute to prosody prediction on the same database.

The two ML-based methods did not differ significantly in the perceptual evaluation. Thus, it can be concluded that the symbolic level of prosody prediction can be safely skipped. On the other hand, the inclusion of symbolic prosody prediction is not detrimental either. The error introduced by symbolic prosody prediction perceptually equals the amount of error produced by the direct method that does not exploit any symbolic prosody features. Therefore, the decision whether or not to include the symbolic prediction can be based entirely on the purpose of the synthesis system. If it is an instructional or research tool (such as MARY), one should include the symbolic prediction level, if it is just a “black box” for the user, one can use the direct prediction method.

As a general rule, the more experienced a TTS user, the higher his or her expectations regarding naturalness. If we aim for a wider usage of speech synthesis, it is necessary to further improve it. More time and effort could be spent introducing other features and trying out different machine learning methods. However, it is doubtful whether the resulting models would lead to a perceptually improved output. The limitations of the KCoRS and MBROLA might have been reached with the presented approach.

One major drawback of the KCoRS is its textual material consisting almost entirely of isolated sentences. In order to model prosodic properties of longer texts, a corpus of read newspaper texts or radio news should be exploited. An even more promising approach is to try a different synthesis method, namely non-uniform unit selection, which generally produces more natural sounding output. However, the available speech material per speaker in the KCoRS is not sufficient for a reliable unit selection speech synthesiser. Therefore, it would be worthwhile to produce such a large labelled speech corpus for German. With this corpus of read speech, one could also include breathing pauses occurring in read speech, making the generated output sound more natural.

References

Baayen, R. Harald; Piepenbrock, Richard; Gulikers, Léon 1995. The CELEX Lexical Database (Release 2). CD-ROM: Linguistic Data Consortium, University of Pennsylvania, Philadelphia, USA [Distributor]

Brants, Thorsten 2000. TnT – a statistical part-of-speech tagger. In: *Proc. ANLP-2000*, Seattle, USA

Brinckmann, Caren 2004. The 'Kiel Corpus of Read Speech' as a resource for speech synthesis. *Master's thesis*: Saarland University: Saarbrücken, Germany. Retrieved February 28, 2005, from http://www.brinckmann.de/KaRS/

Brinckmann, Caren; Trouvain, Jürgen 2003. The role of duration models and symbolic representation for timing in synthetic speech. In: *International Journal of Speech Technology* **6**(**1**), 21–31

Declerck, Thierry 2002. A set of tools for integrating linguistic and non-linguistic information. In: *Proc. SAAKM 2002*, Lyon, France

Dutoit, Thierry; Pagel, Vincent; Pierret, Nicolas; Bataille, François; van der Vrecken, Olivier 1996. The MBROLA project: Towards a set of high quality speech synthesizers free of use for non commercial purposes. In: *Proc. ICSLP 96*: Vol. 3, Philadelphia, USA. 1393–1396

Fackrell, Justin; Vereecken, Halewijn; Martens, Jean-Pierre; Van Coile, Bert 1999. Multilingual prosody modelling using cascades of regression trees and neural networks. In: *Proc. EUROSPEECH '99*: Vol. 4, Budapest, Hungary. 1835–1838

IPDS 1994. The Kiel Corpus of Read Speech. Volume I. CD-ROM: Universität Kiel, Germany

ITU-T 1996. Methods for subjective determination of transmission quality. ITU-T Recommendation P.800: International Telecommunication Union – Telecommunication Standardization Sector: Geneva, Switzerland

Schröder, Marc; Trouvain, Jürgen 2003. The German text-to-speech synthesis system MARY: A tool for research, development and teaching. In: *International Journal of Speech Technology* **6**, 365–377

Skut, Wojciech; Brants, Thorsten 1998. Chunk tagger – statistical recognition of noun phrases. In: *Proc. ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing*, Saarbrücken, Germany

CAREN BRINCKMANN is a research associate at the Institute of Phonetics, Saarland University (Germany). In different projects she contributed to the evaluation and improvement of German speech synthesis systems, multilingual analysis of speech rhythm, and annotation of corpora for investigating information structure. For her master's degrees in computational linguistics and phonetics she focussed on corpus-based prosody prediction for speech synthesis. She offers courses on speech synthesis and programming. Her current research interests include annotation, analysis and modelling of spontaneous speech and corpus-based conversational speech synthesis. E-mail: caren@brinckmann.de.

MORFESSOR AND HUTMEGS: UNSUPERVISED MORPHEME SEGMENTATION FOR HIGHLY-INFLECTING AND COMPOUNDING LANGUAGES

Mathias Creutz[1], Krista Lagus[1], Krister Lindén[1,2], Sami Virpioja[1]
[1] Helsinki University of Technology (Finland)
[2] University of Helsinki (Finland)

Abstract

In this work, we announce the *Morfessor* 1.0 software package, which is a program that takes as input a corpus of raw text and produces a segmentation of the word forms observed in the text. The segmentation obtained often resembles a linguistic morpheme segmentation. In addition, we briefly describe the *Hutmegs* package, also publicly available for research purposes. Hutmegs contains semi-automatically produced correct, or gold-standard, morpheme segmentations for a large number of Finnish and English word forms. One easy way for the reader to familiarize himself with our work is to test the *demonstration* program on our Internet site. The demo shows how Morfessor segments words that the user types in.

Keywords: unsupervised morpheme segmentation, morphology discovery and induction, language-independent, gold-standard, public resources, demo, Finnish, English

1. Introduction

In the theory of linguistic morphology, morphemes are considered to be the smallest meaning-bearing elements of language. Any word form can be expressed as a combination of morphemes, as for instance the following English words: ‘arrange+ment+s, foot+print, mathematic+ian+’s, un+fail+ing+ly’.

It seems that automated morphological analysis would be beneficial for many natural language applications dealing with large vocabularies, such as speech recognition and machine translation. Many existing applications make use of words as vocabulary units. However, for highly-inflecting languages, e.g., Finnish, Turkish, and Estonian, this is infeasible, as the number of possible word forms is very high. The same applies (possibly less drastically) to compounding languages, e.g., German, Swedish, and Greek.

There exist morphological analyzers designed by experts for some languages (e.g., based on the two-level morphology methodology). However, expert knowledge and labor are expensive. Analyzers must be built separately for each language, and the analyzers must be updated on a continuous basis in order to cope with language change (mainly the emergence of new words and their inflections).

As an alternative to the hand-made systems there exist algorithms that work in an unsupervised manner and autonomously discover morpheme segmentations for the words in unannotated text corpora. *Morfessor* is a general model for the unsupervised induction of a simple morphology from raw text data. Morfessor has been designed to cope with languages having predominantly a concatenative morphology and where the number of morphemes per word can vary much and is not known in advance. This distinguishes Morfessor from resembling models, e.g., (Goldsmith 2001), which assume that words consist of one stem possibly followed by a suffix and possibly preceded by a prefix.

In this work, we present the publicly available Morfessor 1.0 software package. The program segments the word forms in its input into morpheme-like units (see Section 2). For evaluating the segmentation produced by Morfessor or some other segmentation algorithm, we provide the *Hutmegs* package. Hutmegs contains linguistic morpheme segmentations for a large number of Finnish and English word forms, as well as a set of tools for comparing the segmentation proposed by the splitting algorithm (e.g., Morfessor) to the correct segmentation of Hutmegs (see Section 3). The Morfessor and Hutmegs package are available on our Internet web page (`http://www.cis.hut.fi/projects/morpho`) together with an online demonstration program.

2. Morpheme segmentation with Morfessor

Morfessor is a general model framework for unsupervised morphology discovery. It takes as input an unannotated text corpus and produces a segmentation of every word in the corpus. We call the proposed segments *morphs*. The boundaries between the discovered morphs often coincide with linguistic morpheme boundaries.

The Morfessor method works in an unsupervised manner, which means that no linguistic knowledge is preprogrammed into it, except for some very general assumptions about model structure. By observing the language data alone Morfessor comes up with a model that captures regularities within the set of observed word forms. The underlying idea is to find the optimal *morph lexicon*, for producing a segmentation of the corpus, i.e., a vocabulary of morphs that is concise, and moreover gives a concise representation for the corpus. This objective corresponds to Occam's razor, which says that among equally performing models one should prefer the smallest one. A mathematical formulation can be obtained using the Minimum Description Length (MDL) principle or probabilistically using maximum a posteriori (MAP) estimation.

Specific models presented by us can be seen as instances of the general Morfessor family. In this context we call the models as follows: Morfessor *Baseline* (Creutz and Lagus 2002), Morfessor *Baseline-Length* (Creutz and Lagus 2002), Morfessor *Categories-ML* (Creutz and Lagus 2004), and Morfessor *Categories-MAP* (Creutz and Lagus 2005a).

Table 1 shows example segmentations obtained by three of the models for the Finnish words 'megatähdeksi' ("[become a] megastar") and 'megatähdistä' ("from megastars") as well as the English words 'tyrannizes' and 'tyrannizing'. The algorithms produce different amounts of information: the Baseline and Baseline-Length methods only produce a segmentation of the words, whereas the category algorithms (Categories-ML and Categories-MAP) also indicate whether a segment functions as a prefix, stem, or suffix. Additionally, the morph lexicon learned by Categories-MAP contains hierarchical representations for some of its entries. These have been visualized using nested brackets.

Table 1: Examples of word segmentations learned by versions of Morfessor from a 16 million word Finnish corpus and a 12 million word English corpus. Proposed prefixes are underlined, stems are rendered in **bold-face**, and suffixes are *slanted*. Square brackets [ ] indicate higher-level entries in the hierarchical lexicon learned by Categories-MAP.

Baseline	Categories-ML	Categories-MAP
mega tähdeksi	mega **tähd** *e ksi*	[**mega** [**tähde** *ksi*]]
mega tähdistä	mega **tähd** *i stä*	[**mega** [**tähdi** *stä*]]
tyrann ize s	**tyrann** *ize s*	**tyrannize** *s*
tyrann izing	**tyrann** *izing*	**tyranni zing**

2.1. Software

The Morfessor 1.0 software package is publicly available on the Internet. The software consists of a Perl script and it is documented in a technical report (Creutz and Lagus 2005b). The Morfessor 1.0 package implements the Morfessor Baseline and Baseline-Length methods.

2.2. Internet demonstration

In addition to the downloadable software, there is a demonstration program on our Internet site. The user types in words of his own choice and the demo shows the analysis (segmentation) that Morfessor produces for these words. It is possible to select the model (Baseline or Categories-ML) and the data used for training the model. Small and large Finnish and English corpora are available.

3. Evaluation of the segmentation with Hutmegs

The Helsinki University of Technology Morphological Evaluation Gold Standard (Hutmegs) package contains fairly accurate morpheme segmentations for 1.4 million Finnish and 120 000 distinct English word forms. To produce these gold-standard segmentations for the words, we have processed the output of the two-level morphology analyzer FINTWOL (Koskenniemi 1983) and the contents of the English CELEX database (Baayen et al. 1995). For every word, an alignment between a surface (or allomorph) segmentation and a deep-level (or morpheme) segmentation has been obtained, as in the following examples:

```
megatähdeksi      mega:mega|PFX tähd^e:tähti|N ksi:TRA
megatähdistä      mega:mega|PFX tähd:tähti|N i:PL stä:ELA

tyrannizes        tyrann:tyrant|N iz^e:ize|s s:V+e3S
tyrannizing       tyrann:tyrant|N iz:ize|s ing:V+pe
```

For instance, the surface segmentation of the English word ‘tyrannizing’ is ‘tyrann + iz + ing’, which has the underlying deep-level representation ‘tyrant + ize + V+pe’. (Here, the label ‘V+pe’ corresponds to the present tense participle ‘ing’.) Additionally we know that ‘tyrant’ is a noun (N) and that ‘ize’ is a suffix (s).

There is also an option for so called “fuzzy” boundaries in the Hutmegs annotations (marked with ^). Fuzzy boundaries are applied in cases where it is inconvenient to define one exact transition point between two morphemes. For instance, in English, the final ‘e’ is dropped in some forms. Here we allow two correct segmentations, namely the traditional linguistic segmentation in ‘tyrann + ize + s’ as well as the alternative interpretation, where the ‘e’ is considered part of the following suffix: ‘tyrann + iz + es’. (The latter can be compared to the form ‘tyrann + iz + ing’, where there is no ‘e’.) The forms of the Finnish noun ‘tähti’ (star) behave in a similar way: In singular forms, we allow the final ‘e’ to belong to the stem or the ending: ‘tähde + ksi’ vs. ‘tähd + eksi’; in plural there is no stem-final vowel and the segmentation is always ‘tähd + i + stä’.

3.1. Experiment

Hutmegs also contains a number of Perl scripts and Makefiles for performing a quantitative evaluation of some suggested segmentation in relation to the desired segmentation in the Gold Standard. In Figure 1 we have plotted the results of an experiment, where the Morfessor Baseline model has been applied to different sized Finnish and English data.

Figure 1a shows how the *precision* and *recall* of the discovered morpheme boundaries develop, when the amount of data increases. Precision is the proportion of correctly discovered boundaries among all boundaries discovered by the algorithm. Recall is the proportion of correctly discovered boundaries among all correct boundaries. In order to get a comprehensive idea of the performance of a method, both measures must be taken into account.

A measure that combines precision and recall is the *F-measure*, which is the harmonic mean of the two and allows for a direct comparison of the goodness of segmentations. Figure 1b depicts the F-measure as a function of the data size.

Generally, the behavior of the Morfessor Baseline algorithm is such that precision increases as a function of the data size. That is, the proposed morpheme boundaries coincide more and more with morpheme boundaries in the Gold Standard. However, when the amount of data is very large, recall starts to decrease. That is, an increasing number of morpheme boundaries in the Gold Standard are missed by Morfessor Baseline.

3.2. Access

The Hutmegs package is a collection of files and documentation that are free to use for non-commercial purposes. However, to obtain the complete Finnish Gold Standard, a missing component must be licensed from Lingsoft, Inc. at an inexpensive price[1]. If the component is not purchased, the user will have access to all Hutmegs scripts and documentation, but only a sample Gold Standard containing the analyses of 700 Finnish word forms.

Likewise, the CELEX database is a prerequisite for accessing the complete English Gold Standard. Non-commercial licenses are available from the Linguistic Data Consortium[2]. The Hutmegs package provides sample segmentations for roughly 600 English word forms, which can be viewed without access to the CELEX database.

More detailed information about Hutmegs is available in a technical report (Creutz and Lindén 2004).

4. Conclusions

Currently, only the Baseline and Baseline-Length versions of Morfessor exist as public resources. The later models (Morfessor Categories-ML and Categories-MAP) may be

[1] URL: `http://www.lingsoft.fi`. Current price: 600 euros.

[2] URL: `http://www.ldc.upenn.edu/`. Current price: US$ 150.

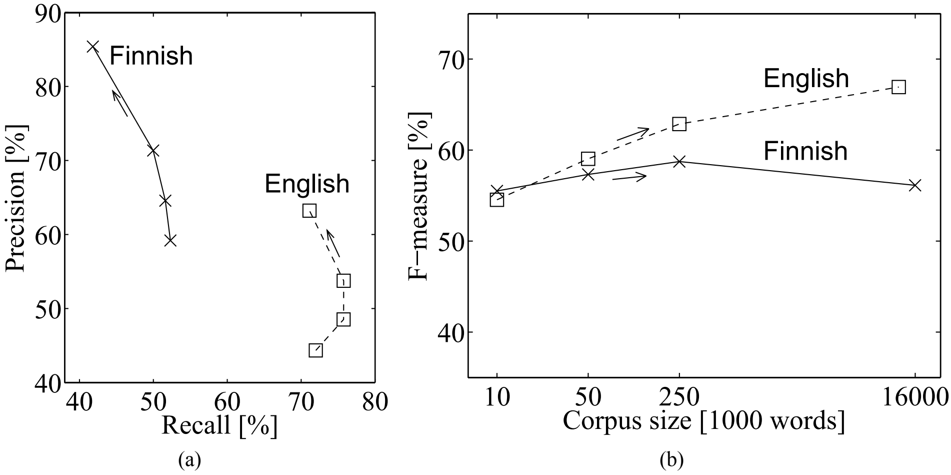


Figure 1: Performance of the Morfessor Baseline algorithm when evaluated against the Hutmegs Gold Standard on different sized corpora of Finnish and English text. In (a) the precision and recall of the discovered morpheme boundaries are shown. The points on the curves correspond to different data sizes and arrows indicate the direction of increasing data size. Precision measures the accuracy of the proposed splitting points, whereas recall describes the coverage of the splits. The most desirable area of the curve is the upper right corner, where both precision and recall are high. In (b) the corresponding F-measure values are shown as a function of the corpus size. The F-measure for Finnish is fairly constant across the corpus sizes, whereas the goodness of the English segmentation seems to improve when increasing the amount of data from 10 000 to 12 million words.

released in the future.

The segmentations in the Hutmegs Gold Standard have been designed for evaluating the accuracy of an unsupervised morphology-discovery algorithm. However, the given morpheme segmentations can also be used for other purposes. For instance, one can estimate n-gram language models from a corpus, where the words have been split into morphemes according to the Gold Standard. Such a language model can be utilized in unlimited-vocabulary continuous speech recognition; see e.g., (Siivola et al. 2003).

In conclusion, by supplying public benchmarking resources, we wish to contribute to the promotion of research in the fascinating field of unsupervised morphology discovery and morpheme segmentation.

References

Baayen, R. Harald; Piepenbrock, Richard; Gulikers, Léon 1995. The CELEX lexical database (CD-ROM). University of Pennsylvania, Philadelphia, PA: Linguistic Data Consortium. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC96L14`

Creutz, M.; Lagus, K. 2002. Unsupervised discovery of morphemes. In: *Proc. Workshop on Morphological and Phonological Learning of ACL'02*, Philadelphia, Pennsylvania, USA. 21–30

Creutz, Mathias; Lagus, Krista 2004. Induction of a simple morphology for highly-inflecting languages. In: *Proc. 7th Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, Barcelona. 43–51

Creutz, Mathias; Lagus, Krista 2005a. Inducing the morphological lexicon of a natural language from unannotated text. In: *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*. (submitted for review)

Creutz, Mathias; Lagus, Krista 2005b. Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor. Technical Report A81: Publications in Computer and Information Science, Helsinki University of Technology

Creutz, Mathias; Lindén, Krister 2004. Morpheme Segmentation Gold Standards for Finnish and English. Technical Report A77: Publications in Computer and Information Science, Helsinki University of Technology

Goldsmith, John 2001. Unsupervised learning of the morphology of a natural language. In: *Computational Linguistics* 27(2), 153–198

Koskenniemi, K. 1983. *Ph.D. thesis*: Two-level morphology: A general computational model for word-form recognition and production, University of Helsinki

Siivola, V.; Hirsimäki, T.; Creutz, M.; Kurimo, M. 2003. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In: *Proc. Eurospeech'03*, Geneva, Switzerland. 2293–2296

MATHIAS CREUTZ is a post-graduate researcher at the Neural Networks Research Centre, Helsinki University of Technology (HUT). He received his M. Sc. degree at HUT in 2000 and is now working on his Doctoral thesis. His research interests concern the unsupervised induction of a morphology of natural languages, the automatic segmentation of words, and the application of subword-unit-based language models in unlimited vocabulary speech recognition. E-mail: mathias.creutz@hut.fi

KRISTA LAGUS is a teaching research scientist at the Neural Networks Research Centre, Helsinki University of Technology. She received her Ph.D. at HUT in 2000, dealing with text mining using the WEBSOM neural network method. Her research interests concern using machine learning methods for modeling the emergence of representations of language and cognition in artificial systems. She has taught courses on statistical language modeling and related topics at HUT. She has published 6 articles in international journals, several book chapters and over 30 conference articles. E-mail: krista.lagus@hut.fi

KRISTER LINDÉN is a postgraduate researcher at the Department of General Linguistics, Helsinki. He received his M.Sc. at the University of Helsinki. His research interests concern word sense disambiguation and word sense discovery and their application to machine translation, speech recognition and cross-lingual information retrieval. E-mail: krister.linden@helsinki.fi

SAMI VIRPIOJA is an undergraduate researcher at the Neural Networks Research Centre, Helsinki University of Technology. He is currently working on his Master's thesis concerning natural language modeling. E-mail: sami.virpioja@hut.fi

FOREIGN LANGUAGE READING TOOL – FIRST STEP TOWARDS ENGLISH-LATVIAN COMMERCIAL MACHINE TRANSLATION SYSTEM

Daiga Deksne, Inguna Skadiņa, Raivis Skadiņš, Andrejs Vasiļjevs
(Tilde, Riga, Latvia)

Abstract

Foreign language reading tool is a software technology developed by Tilde to facilitate Latvian users in reading English web pages and other onscreen documents. It includes the following constituents: English parser, English-Latvian transfer rules, English-Latvian translation dictionary, Latvian disambiguation tool and Latvian phrase/sentence generator. The transfer rules, disambiguation tool and Latvian phrase generator are new features developed for the tool. The task of transfer rules is to transform English syntax structures into corresponding Latvian syntax structures. For instance, an English noun phrase with the preposition *of* is translated into Latvian as a genitive noun phrase: noun in genitive+head noun. Since most of words in text are ambiguous or have more than one translation into target language, the disambiguation tool is developed. The task of the disambiguation tool is to choose the most probable translation between all theoretically possible translations generated by word-to-word translation. The third new constituent is the Latvian phrase generator. It is used to generate syntactically correct Latvian sentence fragments which are very important for inflective languages like Latvian. This is a rule-based constituent and is closely related to transfer rules.

Keywords: translation, machine translation, context, morphology, parsing, disambiguation, idioms, phrase

1. Why Reading Assistant instead of Machine Translation

In the world, Machine Translation (MT) systems are very popular. However, most of them are criticized by their insufficient quality. Another approach was proposed by Gábor Prószéky and Kis Balázs (Prószéky, Balázs 2002): instead of an automated MT system, most of users are satisfied with reading assistant which allows understanding of text. Studies of user habits and foreign language knowledge of Latvian users showed us that such tool could be useful and demanded in Latvia also.

Users who have some knowledge of English do not want to trust Machine Translation system, because it is not perfect. Instead of that they prefer to read original text and use assistance only when it is necessary. Users want the assistant to help them understand complicated parts of the text. In these complicated parts of the text the system will help them:

- to understand sentence or phrase structure,
- to find relations between words,

- to identify idiomatic meaning,
- to provide possible translations of phrase and of each word in context.

User appreciates freedom to interpret text himself instead of receiving only one translation as Machine Translation systems usually offer.

The developed reading assistant tracks mouse pointer and retrieves text under it. Then system analyzes the text, finds translation of the phrase containing the word under cursor and finds all translations for each word in phrase. Then all results are presented to the user. Currently, reading assistant can retrieve and translate texts from Microsoft Office documents, web pages in Internet Explorer, Windows and applications menus, toolbars, dialogs and controls.

2. The Structure of the Translation System

The aim of the translation system is to identify individual phrases in the text and provide user with the full translation of the whole phrase, as well as separate translations of the words constituting the phrase. If the system cannot identify a phrase, the translations of individual words are provided. The translation system is built from separate components each of them having their own functionality. (See Figure 1). The components operate successively. It means that the first component receives input data: the text to be translated and the position of the word over which the cursor is placed in the text. The position data of the current word are required in order to know which phrase translation should be returned to the user if the input text is translated to more than one phrase. The input data for each subsequent component are the internal structures created or processed by the previous component. The last component in the chain returns ready translation to the user. It contains both the phrase translations and the translations of each word of the phrase.

(1)

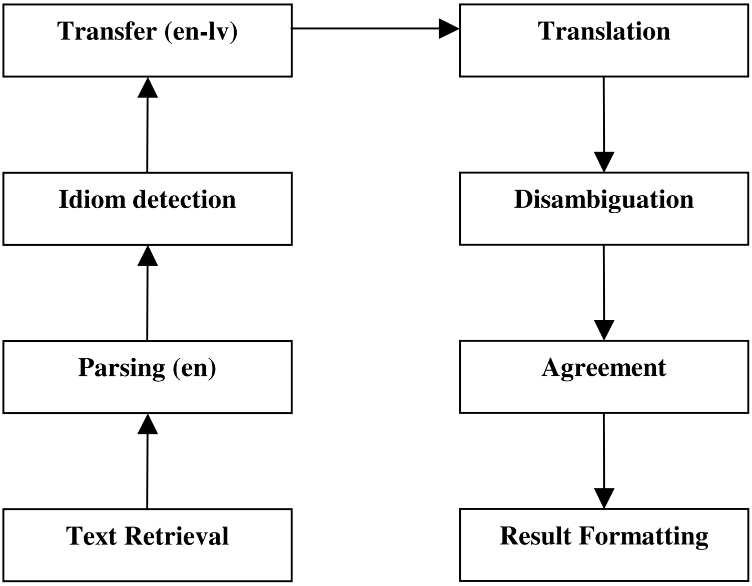


Figure 1. The chain of the translation system components

The aim of the text retrieval component is to obtain clear text in a form understandable for the parser. The text is divided into sentences and placed in internal structures remembering the position of the word in the original text.

The aim of the parser component is to obtain a fully or partially parsed sentence. The parser component creates the syntactic tree of the sentence. The relations included in the tree are very diverse. The relation is determined by the class of the primary word, the class and the type of relation of the dependant word. There may be different types of relations between different classes of words. For example, between a verb and a noun there may be an *object* or *subject* type relation, between an adjective and a noun — an *attributive* type relation, etc.

Very often in texts there are idiomatic phrases and their meaning differs from the meaning of the individual words forming these phrases, and there is no directly matching phrase in Latvian. The example (2 a) provides such idiomatic phrase, (2 b) is the Latvian translation of this idiom, (2 c) the literal English translation of the corresponding Latvian idiom. As we see, (2 a) and (2 c) are completely different phrases. Example (3 a-c) provides another similar case. So, there is no sense to translate individual words; we have to translate the phrase as a whole. Such idioms are identified successively while attempting to translate adjacent words in the text. The whole sentence syntactic tree information is not used in finding and translating idioms, however, the translated idiom is integrated into the tree to use it later in transfer, agreement and other processes.

(2) a. It rains cats and dogs.
b. Gāž kā ar spaiņiem.
c. Pouring like with buckets.

(3) a. A fly in the ointment.
b. Piliens darvas medus mucā.
c. A drop of tar in the barrel of honey.

The tool also translates interface elements of the Windows operating system and software applications, not only texts. Mostly, user interface texts cannot be translated literally and it is important to use correct terminology in translation. Therefore, user interface texts are translated using another, specific dictionary of idioms and terminology, allowing to get more accurate translations of MS Office and MS Windows user interface elements.

The task of the transfer component is to convert English syntactic trees into corresponding Latvian syntactic trees. Not always it is possible due to the free word order allowed in Latvian. The transfer component uses a set of rules that describes how specific relations in English syntactic tree are converted into relations of the corresponding Latvian syntactic tree. Transfer rules may change word order, transfer or assign properties, change type of word relations. In the rule example (4) relations for phrase 'team of scientists' are changed. The word order is changed, the describing word 'scientists' is moved to the position before the main word 'team', the case of the word is changed to the possessive case, preposition 'of' is discarded.

```
(4) TransferRule(N<-modPREP<-pcomp-N)
{
     Child.SourceSpelling == "of";
     Grandchild.Case = genitive;
     MakeLink(Child – hidden -> Parent);
     Swap(GrandChild, Parent);
     MakeLink(GrandChild - mod -> Parent);
 }
```

There are many groups of transfer rules — rules that process perfect tenses of verbs (5 a), negative verbs (5 b), relations of subjects and verbs (5 c), relations of verbs and objects (5 d), degrees of adjectives (5 e), relations of attributes and nouns (5 f), connections of attributes with conjunctions (5 g), words connected to prepositions (5 h), verbs with postpositions (5 i) and other cases.

(5) a. ... has been translated
b. ... did not write
c. ... boy writes ...
d. ... writes ... letter
e. most significant
f. consistent terminology
g. consistent and ...
h. in ... park
i. ... tracked ... down

The translation component selects the translation based on the class of the word identified by the parser component for the current word in the sentence. In English, a word often can be both a noun and a verb. During parsing of the sentence this ambiguity is eliminated. If there is no translation for the word in the required class, the translation is attempted for alternate classes, for instance, instead of a participle the translation of an adjective can be required. Usually, dictionaries include only translations of primary words without translations of words easily and regularly derived from the primary words. For example, dictionaries usually have entries for words like ‘assume’, but less often they have entries for ‘assumption’, ‘assumed’ (adverb) or ‘assuming’ (noun) and they usually do not have entries for words like ‘assumer’ and ‘assumingly’. If the translation of a word cannot be obtained, specific suffixes and prefixes are cut off the end and beginning of the word. For example, a participle can be translated as the infinitive of the corresponding verb and then, the required participle form synthesized from the translation. The translations are arranged by their significance. Each translation has a label attached identifying whether it can be used in translation of the phrase. Too specific translations are not used in forming the phrase, they appear only in the list for each word.

Since the translation component usually returns more than one translation for a word, the task of the disambiguation component is to select the most appropriate translation to apply in the phrase. All components described above are rule-based, but in disambiguation we use statistical approach. As a result of processing a large Latvian text corpus, a data base was created containing probabilities between various pairs of words and various types of relations. This database is used to calculate the common probabilities of translations of words of one phrase. In beginning of disambiguation process, we know all possible translations for each word, we know syntactic relations they are in, and we know probabilities for each pair of words to be in particular relation. In disambiguation process we select those translations which give highest probability for the whole phrase.

At the end of disambiguation process we have Latvian syntactic tree with only one Latvian word in each tree node. Tree nodes also have some morphological properties set, for example, verbs have property of tense, nouns have property of case and number. But not all properties are set. There are properties which we can get from English text, like tense of verb or number of noun. Other properties are determined by syntax laws, for example, subject must be in nominative case and object must be in accusative case. Other properties we can get from dictionary, for example, gender of noun. But there are properties which must be set depending on properties of other words. For example, in Latvian noun and adjective must agree in case, number and gender, verb must agree with subject in person etc. This agreement is established by special Agreement module. This module passes attributes from one word to other and sets missing morphological properties so that all morphological properties are set and all words in phrase are in agreement. We can look at this module also as Latvian text

Generation module, because it takes in words, their relations and some attributes and transforms that in syntactically correct Latvian phrase.
The last phase in process chain is formatting of results. Result Formatting module has all tree structures of input sentence or paragraph and it knows in which word in sentence user is interested. Module finds largest English phrase containing this word, finds corresponding Latvian phrase and finds all translations for all English words in English phrase. And finally results are presented to the user.

3. Achieved results and future work

Our approach for separate text fragment translation is easy-to-use and helpful for the user. Quality of translation of phrases varies depending on complexity of the text. System can handle relatively simple phrases, but fails dealing with texts from specific domains or dealing with texts with complex grammar and idiomatic meaning, like news headlines. We have done some testing and evaluation of the system, and we have discovered several weaknesses of the system. This is the basis for future work on improvement of the system.

During parsing of text, proper nouns are not distinguished, therefore, they sometimes are translated with a standard dictionary and the obtained translation does not match the context. In future, we should improve proper noun recognition functionality and they should be identified and translated using special dictionaries.

There are problems with noun phrases with two nouns. In English phrases, two word noun phrases can be formed in several ways, for example, *team of scientists*, *scientists' team*, *language technology* and *teacher John*. In Latvian, first three phrases must have the first word in genitive case, but in the fourth phrase both words must agree in case. Currently, the English parser we use would not make any distinction between the last three phrases; it may result in incorrect case for translation.

There is still a lot of work to be done to improve the quality of the dictionary. Our translation system assumes that translations in dictionary are ordered so that most common translations go first and only then go rare and specific translations. We want to use common translations with higher probability in phrase translation and we want to use specific translation with lower probability or do not use them at all. Frequently, in contemporary dictionaries words which are outdated or are not often used in texts are given as the first meaning. To improve translation quality revised dictionary is necessary which would meet usage-specific criteria. Ordinary contemporary dictionaries are created to be good reference source; they have a lot of information and provide as exhaustive description of the word as possible. For automated translation used in our system different approach is necessary — we need strict distinction between common translations and specific translations used in specific context.

Quality of a dictionary is closely related with the quality of the disambiguation process. If we have a lot of translations for all words and these translations do not have correct sorting order in the dictionary, it is obvious that we get strange disambiguation results, because the disambiguator tries to find the most probable combination of translations and it can take two very specific translations and find that they fit together better than others. Currently, disambiguator is trained only on monolingual shallow parsed target language corpora. We could improve disambiguator results by training it on aligned parallel corpus. But unfortunately, there is no large-scale parallel aligned English-Latvian corpora available.

References

Prószéky Gábor, Balázs Kis 2002. Development of a Context-Sensitive Electronic Dictionary. In: Braasch, Anna and Povlsen, Claus (eds.). *Proceedings of the Tenth EURALEX International Congress, EURALEX 2002*. Copenhagen, Denmark. 281–290.

INGUNA SKADIŅA, Dr Comp. Sc. Doctor's degree with the thesis "Latvian Language Modelling for Artificial Intelligence Systems" in 1997. In 1999, I. Skadiņa joined Tilde's Language technology group. Since 1989 I. Skadiņa is cooperating with the Artifical Intelligence Laboratory of the Institute of Mathematics and Computer Science, the University of Latvia. Her recent research interests are concerned with computational syntax, semantics and machine translation.

RAIVIS SKADIŅŠ graduated from the University of Latvia in 1994. He is working in Tilde since 1995. Raivis Skadiņš has worked on development of the automated English-Latvian-English dictionary, developed Latvian spellchecker, thesaurus and hyphenator. Since 1998 he is the Head of the Language&Reference System group in Tilde. In 2001, Raivis Skadiņš received Master's Degree in Computer Science. His master thesis "Object-oriented analysis and Universal Networking Language (UNL)" investigates UNL and object-oriented analysis and studies formal means they offer to describe the real world.

DAIGA DEKSNE graduated from the University of Latvia in 1991. She is working at software company Tilde since 1997. Since 2004 she is leading the development team which develops all language technology products at Tilde.

ANDREJS VASIĻJEVS graduated from the University of Latvia in 1992. In 1996, he received Master's Degree in Computer Science. In 1991, Andrejs Vasiļjevs was one of the founders of the Baltic software company Tilde. He is director of software development and responsible for products and services developed at Tilde. Andrejs Vasiļjevs is a member of the Board of the Latvian Information Technology and Telecommunications Association, Soros Foundation Latvia. He is a member of the Commission of Official Language responsible for development of HLT.

SUB-BAND OVERLAP-ADD TIME-SCALING OF SPEECH IN BACKGROUND NOISES

Mike Demol[*], Werner Verhelst[*], Kris Struyve[**], Piet Verhoeve[**]
*Laboratory for Speech and Audio Processing, dept. ETRO-DSSP, Interdisciplinary Institute for Broadband Technology, Vrije Universiteit Brussel, Belgium
**Central R&D Department, TELEVIC nv, Izegem, Belgium

Abstract

Time domain time-scaling algorithms like WSOLA are capable of delivering high quality results by relying on the noisy or quasi periodic nature of speech segments. Such simple structures are not guaranteed for more complex audio signals like music. Sub-band WSOLA was developed as an extension of WSOLA for time-scaling such polyphonic signals. In this paper we investigate the use of this sub-band approach for speech corrupted by background noises. We propose a rule of thumb to determine the number of sub-bands and optimize the timing tolerance parameters of the different sub-bands. Sub-band WSOLA is compared with full-band WSOLA through informal listening tests.

Keywords: Robust time–scaling, WSOLA, sub-band WSOLA, speech modification

1. Introduction

Synchronized overlap-add techniques are simple and efficient techniques for time-scaling of speech (Verhelst 2000). We introduced waveform similarity based overlap-add (WSOLA) based on the notion that a time-warped version of an acoustic signal should be perceived to consist of the same acoustic events as the original signal, but with a modified timing structure. In (Verhelst et al. 1993) we assumed that this would be the case if the waveform of the time-warped signal is maximally similar to the waveform of the original signal in all neighbourhoods of corresponding time indices.

The criterion of waveform similarity proved a good substitute for the criterion of sound similarity if the similarity can be kept very close. This is the case for quasi-periodic signals and quasi-stationary noise (hence, also for clean speech), but not for more complex structured audio signals. In (Spleesters et al. 1994), we developed a refined system based on the idea that proper sound similarity will be achieved if the time-warped signal produces a neural firing pattern that is maximally similar to the original firing pattern in all neighbourhoods of corresponding time instants. In that system, the input signal is passed through a perfect reconstructing filter bank, and the resulting sub-band signals are considered characteristic for the auditory nerves firing pattern (center frequency for the bundle of neurons excited, and power for the firing rate). By applying WSOLA to each of the sub-band signals separately, we thus approximated local similarity of firing patterns.

In the present paper, we investigate the performance of sub-band WSOLA for the specific case of speech signals corrupted by music, background noise, etc. It is shown that excellent results can be obtained for this particular type of signals by taking advantage of the fact that most of the harmonically distributed energy in speech resides in the lower frequency region.

In Section 2 we briefly review sub-band WSOLA time-scaling. In section 3 we discuss the application of sub-band WSOLA for the particular case of speech in background noise. In section 4 we describe our experiments and discuss our results. Finally, in section 5 we conclude the paper and refer to possibilities for future work.

2. Sub-band waveform similarity overlap-add

In its basic form, the overlap-add (OLA) strategy for time scaling consists of excising segments at time instants $\tau^{-1}(L_k)$ from the input signal $x(n)$, shifting them to time instants L_k, and adding them together to form the time scaled output signal $y(n)$[1]:

$$y(n) = \Sigma_k\, x(n + \tau^{-1}(L_k) - L_k)w(n - L_k).$$

In constructing the output signal in this way, the individual segments will add incoherently, which introduces structural discontinuities at the waveform segment joins. WSOLA introduces tolerance parameters Δ_k ($\in$ $[-\Delta_{max} .. +\Delta_{max}]$) on the desired time-warping function to ensure that each new output segment $x(n + \Delta_k + \tau^{-1}(L_k) - L_k)\, w(n - L_k)$ can be added coherently to the already formed portion of the time-scaled signal. WSOLA ensures this signal continuity at segment joins by requiring maximal similarity between the new output segment and the segment that followed the previous output segment in the input signal (Verhelst et al. 1993).

The structure of polyphonic signals like music can be fairly complicated with several incoherent pitches. In such situation the timing tolerance Δ does not provide the necessary freedom to ensure signal continuity at segment joins. Figure 1 illustrates a situation with two periodic signals that can perfectly be time-scaled by WSOLA when considered individually. However, no single segmentation could ensure continuity at the segment joins for the sum of these signals

A polyphase perfect reconstruction uniform DFT filter bank was therefore proposed to decompose the original signal in a set of simpler sub-band waveforms, which can be individually time-scaled by WSOLA before constructing the output from the time-warped sub-band signals, see Figure 2. Quality improvements were regularly achieved for musical signals when compared to the full-band application of WSOLA. When the sub-band tolerance parameters Δ were chosen too large, the resulting signal tended to suffer a loss of power and purity on perceived aspects of dynamics and rhythmic. A similar type of distortion occurred when each sub-band signal was delayed by an independent random number of samples before summing all sub-band signals without time-scaling. The loss of inter-band synchronicity thus explains the distortions that come with large sub-band timing tolerances Δ.

[1] Synthesis time instants L_k and the window function $w(n)$ are chosen such that $\Sigma_k\, w(n - L_k) = 1$. In practice we use $L_k = kL$ with 50% overlapping hanning windows for this purpose.

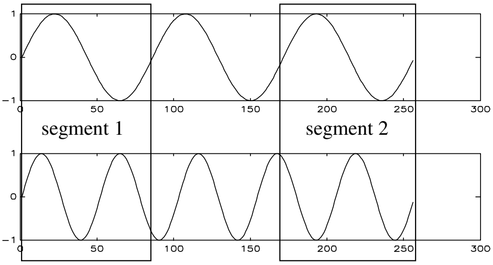


Figure 1: The concatenation of segments 1 and 2 produces a clean low-frequency tone but distorts the high frequency tone

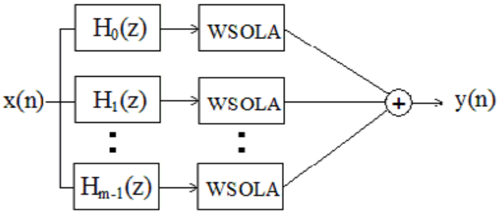


Figure 2: Sub-band WSOLA using an Mth band filter bank

3. Application to speech in noisy environments

In the case of speech corrupted by background noises a similar reasoning is valid: the quasi-periodic structure of clean speech can be blurred by the structure of the noise signals resulting in a complex overall waveform structure. Applying sub-band WSOLA on this type of signal should improve the time-scaling results.

Since we wish the time-scaled voice to be maximally free from synchronization artefacts, we chose the number of sub-bands such that most of the periodic energy of the speech signal is grouped together in one sub-band. For speech the first formant is in the range of 0-1000 Hz and contains most of the periodic energy. Therefore, we chose the cut-off frequency of the first band around 1 kHz. Compared to full-band WSOLA, the SNR in this first band will now be much higher and should allow sub-band WSOLA to better preserve the pitch structure of the signal and to deliver better time-scaled speech. Additionally, the background noise is also expected to be less distorted since the higher sub-band signals are time-scaled separately using the synchronization information from their respective sub-bands.

It can be seen that for proper operation on a periodic signal, the window length of WSOLA should span at least one period and Δ_{max} should be at least half a period. When a periodic signal is passed through a filter bank, the resulting sub-band signals have the same period as the original signal. If this would apply for (voiced) speech, the period of all sub-band signals would be the same and hence we should choose a same window length and tolerance for all the sub-bands. However, the higher sub-bands of speech do not usually show a pitch periodicity, see Figure 3. Since it is computationally more efficient, we therefore chose to determine the window length and timing tolerance in accordance with the longest period of a sinusoidal signal that fits in the sub-band channel considered. Table I shows an example of the cut-off frequencies and the corresponding maximal period for several bands. In the first band we can chose a period of 30 ms instead of ∞ since in voiced speech no harmonic below 33 Hz is expected. For the high-frequency bands, the absolute difference between the periods becomes small and the bands do not carry much speech energy. We therefore summed several high-frequency bands together before time-scaling them. This reduced the number of sub-bands to time-scale to typically 4-5 and has a positive influence on the loss of inter-band synchronicity (fewer timing tolerance parameters Δ). The sub-band window lengths were chosen according to a power law, see Figure 4. The bottom curve represents the true band periods and the top curve the window lengths used for time-scaling each sub-band. The tolerance interval ($2\Delta_{max}$) was chosen equal to the window length.

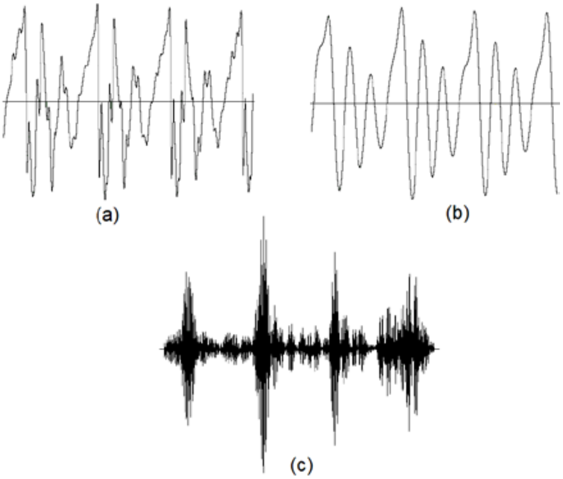


Figure 3: Voiced clean speech segment, sampled at 32 kHz. (a) full-band input signal (b) band 1 (0-1 kHz) (c) band 8 (13-15 kHz)

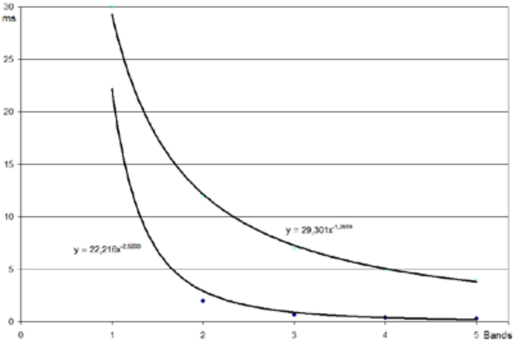


Figure 4: Choice of window lengths for the different sub-bands

Table I: Example sub-band frequencies and corresponding periods

Freq. range (kHz)	0-1	1-3	3-5	5-7	7-9
max. period (ms)	∞ => 30	1	0.333	0.2	0.143

4. Experimental results

The goal of the experiments was to evaluate whether the sub-band approach, initially introduced for musical fragments, could improve the quality of time-scaled speech in background noises. Synchronization artefacts in WSOLA are more easily audible in slowed-down signals. We therefore used moderate slow-down factors in the experiments (typically around 0.7). We selected clean speech fragments to which we mixed music or babble noise at different SNR levels (10, 5, 2, 0 dB) at f_s=22.05 kHz. The audio fragments were time-scaled using full-band and sub-band WSOLA and compared. Quality was evaluated informally by the authors using high-quality consumer headphones.

From the first test results, it appeared that there is a difference between high SNR (10, 5 dB) and low SNR (2, 0 db). At low SNR the typical full-band synchronization artefacts like roughness in the speech or irregularities in the high frequency components of the noise where more disturbing and audible then at high SNR for full-band WSOLA. With sub-band WSOLA these artefacts were absent but a new type of artefact became audible, namely the speech signal was 'smeared out'. This blurring effect was most perceptible at high SNR and less at low SNR. At lower SNR the smearing effect of the speech is also present but less audible due to the masking effect of the noise or background music. In some cases the blurring of the voice could become as disturbing as the synchronization artefacts of full-band WSOLA. This effect is due to the different sub-band timing tolerances Δ, which introduce a loss of inter-band synchronicity. From the test we thus noticed that although we had optimized the number of sub-bands and the different timing tolerances Δ to minimize the effect of inter-band synchronicity loss, it was still significantly present and audible (typically at high SNR).

In a final experiment we wanted to further minimize this loss in inter-band synchronicity. We tested two different variants of the sub-band WSOLA algorithm, in which we chose a same window length for all the sub-bands. In the first version we time-scaled the first sub-band and used the same optimal segmentation (Δ values) for the other bands as well. In the second version we time-scaled the first two bands independently and used the optimal positions of the second band for all the remaining high frequency bands. In this way we could further reduce the number of independent timing tolerance parameters Δ to 1 or 2 and minimize the blurring effect. In the last two versions we actually compromised between the blurring effect and the synchronization artefacts and obtained the best time-scaling results, with a slight preference for the second version over the first version.

5. Conclusion

In this paper we showed that the sub-band WSOLA approach, initially introduced for music, can also be applied for speech in background noises (music or other types of noise). We suggested that for speech the number of sub-bands can be determined by choosing the cut off frequency of the first band to capture most of the periodic energy from the speech (about 1 kHz). The test results showed that it is important that the periodic energy of the speech is processed together in one band. This avoided synchronization artefacts for the speech and since the higher bands were time-scaled independently, the noise components were also properly time-scaled but a blurring of the voice became noticeable. To counter this effect we can apply the timing tolerance parameters Δ of the low sub-bands to the higher sub-bands as well, as a good compromise between the blurring and incorrect synchronization of the speech. To further improve the sub-band approach, some constraints could be set on the timing tolerance parameters Δ for the higher frequency sub-bands. In this way, one could try to further minimize both the blurring of the voice and the synchronization artefacts.

Acknowledgments

The research reported on in this paper was supported by grants from the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT-Vlaanderen).

References

Verhelst, W. 2000. Overlap-Add Methods for Time-Scaling of Speech. In: *Speech Communication*, Vol. 30, No. 4. 207-221

Verhelst, W.; Roelands, M. 1993. An Overlap-Add Technique Based on Waveform Similarity (WSOLA) for High-Quality Time-Scale Modification of Speech. In: *IEEE proceedings of ICASSP-93*. Place: Minneapolis, Minnesota. Vol. 2. 554 - 557

Spleesters, G.; Verhelst, W.; Wahl, A. 1994. On the Application of Automatic Waveform Editing for Time-Warping Digital and Analog Recordings. In: *96th Convention of the Audio Engineering Society*. Place: Amsterdam

MIKE DEMOL received his master degree in electronics at the Vrije Universiteit Brussel in 2004 and is currently a PhD researcher at the department ETRO-DSSP. His research interests are perception based time-scaling of speech signals and speech enhancement. E-mail: midemol@etro.vub.ac.be, Website: www.etro.vub.ac.be/Research/DSSP/dssp.htm

WERNER VERHELST heads the Laboratory for Digital Speech and Audio Processing (DSSP) at the Vrije Universiteit Brussel. He has, a.o., researched speech modification and synthesis at the Institute for Perception Research in Eindhoven (1989-1991), and audio modification and modelling at the Katholieke Universiteit Leuven (1999-2002). E-mail: wverhels@etro.vub.ac.be

KRIS STRUYVE is a Senior Development Engineer at Televic. His present research interests are in digital audio processing and acoustics. His prior research and professional experiences were in the field of broadband telecom networking. He received a PhD degree in electronics from the University Ghent in Belgium. E-mail: k.struyve@televic.com, Website: www.televic.com

PIET VERHOEVE is Sr. Research Associate at Televic. His research activities are in the domain of digital multimedia processing and networks. He received a master degree in electrical engineering at the University of Leuven and a PhD degree in electronics at the University of Ghent. He is author and co-author of several publications and patents. E-mail: p.verhoeve@televic.com

EFFICIENT SEMANTIC PARSING OF CONVERSATIONAL SPEECH

Michel Généreux
Information Technology Research Institute, Brighton, United Kingdom

Abstract

This paper presents an empirical method for mapping speech input to shallow semantic representation. Semantic parsing is realized through a bottom-up type parsing paradigm where the operators are based on semantic concepts, obtained from a lexicon. A statistically trained model specializes the parser, by guiding the runtime beam-like search of possible parses. The semantic representation is a logical form equivalent to a Discourse Representation Structure (DRS). Each output of the parser is given a probability according to how similar, given a contextual word similarity measure, the parsing process for the input was to those collected during the training phase. Contextual information during parsing allows for better coverage of large domains. The non syntactic but very semantic nature of the parser would make it very tolerant to noisy (recognized) speech input. Shallow parsing using First-Order Logic (FOL) allows for fast but meaningful enough processing of the input, which makes the parser well suited for real-time Spoken Dialogs Systems (SDS).

Keywords: Semantics, Corpus, Discourse

1. Introduction and Motivation

For task oriented systems, the quality of the spoken interaction between man and machine have seen constant progress over the last decade. Today, lower word error rate in speech technology, expertise gained in dialog management, more flexible natural language generation and better speech synthesis allows us to take dialog systems to the next level: open (or very broad) domain of interaction. To cope with the complexity of open domain and noisy speech input, semantic parsers will have to put a strong emphasis on context to supplement for syntactical analysis, and outputs a suitable meaning representation for discourse to be further interpreted. We present a parser with strong contextual capabilities that delivers a DRS as output.

The choice of our bottom-up parser is motivated by its manageability and its similarity to how humans parse a sentence [Hermjakob and Mooney (1997)]: compositionally build meaning from left to right by adding concepts as they appear (INTRODUCE), combining them (COREF and DROP operations) and keeping in mind contextual information (SHIFT operation). Finally, FOL [Light and Schubert (1994)] offers a reasonably deep semantic representation and a convenient way to translate DRSs, for carrying out sensible discourse conversations.

2. System Architecture

We propose an approach in which a bottom-up parser similar to [Mooney and Tang (2000)] is combined with a statistical model. Mapping input to logical form is triggered by keywords (INTRODUCE operation) from a semantic lexicon collected from training. Words in the input not in the semantic lexicon are used as contextual information (SHIFT operation).

Three other operations are available to the parser: co-referencing variables (COREF), dropping one term into the argument of another (DROP) and giving scope to quantifiers (SCOPE). In the parser, all operations are conducted within a particular context, a context being a word or group of words following (in the strictly left-to-right sense) a parsing operation. The output of the parser is a a partially resolved DRS, ready to be processed by a Discourse Manager. Figure 1 shows the various elements of the parser.

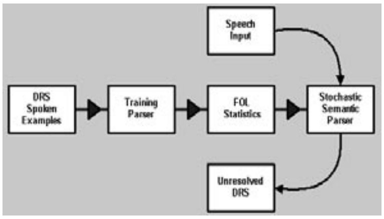


Figure 1: Parser Architecture

3. Overview of the Parsing Process

This section is meant to give a flavor of the parsing process and provides a ”light” introduction to the parser. All statistical considerations are for the moment deferred to section 5. The parser used is a variant of a *Shift-Reduce* parser (only the SHIFT operation has been retained).

The Input String The input string is a list of words to give an interpretation for. When no actions are applicable and the input string is empty, then the parsing process is completed. Typically, one word is removed from the list for a SHIFT action and one or more for an INTRODUCE action. It also provides some contextual information while applying a parsing action. Example 1 shows an input string.

(1) [I,read,The,Little,Mermaid,Did,you,write,it]

The Parse Stack The parse stack is the actual parse state, the current interpretation of the input string found so far. It is a list of binary elements, each element representing a combination of the introduced predicate (or concept) with its context of introduction. The context gives partial (but useful) information on the words following the concept at the time of introduction. Each concept must be in the semantic lexicon. Here is the general format of the parse stack: **[concept1:[context1],concept2:[context2],. . . ,start:[context]]**
The *start* predicate is there only to provide room for words from the input string which would be shifted at the very beginning of the parse; it does NOT contribute to the meaning of the input phrase. The representation of operators, parse state and final state follow.
SHIFT(word_to_be_shifted) A SHIFT action simply puts the first word from the input string at the end of the context of the concept on the top of the parse stack.
INTRODUCE(concept_to_be_introduced) The INTRODUCE action takes a concept from the semantic lexicon (the lookup is triggered by keywords in the input) and puts it on the top of the parse stack, initializing its context of introduction to the word (or list of words) that triggered this concept. These concepts will then participate to the meaning representation.

DROP(source_term, target_term) The DROP action attempts to place a term from the parse stack as argument to another term of the parse stack. The context of the source term is lost in the process. This action has no effect on the input string.
COREF(variable1, variable2) The COREF action attempts to co-reference two variables, in the case at least one of them is underspecified (_). The result is that they become specified (they have the same name). This action has no effect on the input string.
SCOPE(source_term, target_term) The SCOPE action is similar to the DROP action, with the two exceptions that it applies only to quantifiers and that the 'droping' is slightly different. For example, SCOPE(exists(A,human(A)),forall(B,thing(B))) results in the target term being forall(B,exists(A,(human(A),thing(B)))). This action has no effect on the input string.
op(ACTION(arguments)#Parse_Stack#Input_String) indicates in which context, i.e. how the Parse Stack and the Input String looked like, when the action took place. *Op* is simply a container for all types of actions.
final(Parse_Stack) indicates the final aspect of a parse, i.e. the meaning we have found for an input string.

Semantic Lexicon The semantic lexicon comprises all the concepts and their triggering phrase(s) that we wish our parser to process. A triggering phrase is simply a word (or group of words) in the input string that triggers some concept. The format of a lexical entry is: **lexicon(CONCEPT, [TRIGGERING_PHRASE]).**

The bottom-up parser We are now ready to present the variant of the shift-reduce parser we are using. The algorithm of the parser is as follows:

1. Try to INTRODUCE a new concept or SHIFT a word.
2. Do a subset of the following operations {DROP, COREF, SCOPE}.
3. If there are more words in the input string, go back to Step 1. Otherwise stop.

A parsing example We show a complete parsing in the case the user turn is *OOV did you write it?*, where OOV is a out of vocabulary symbol produced by the speech recognizer:

INPUT-OPERATION
*NEW PARSE STACK in FOL
[OOV did you write it]-SHIFT
*[start:[OOV]]
[did you write it]-SHIFT
*[start:[OOV,did]]
[you write it]-INTRODUCE
*[∃(_,system(_)):[you],start:[OOV,did]]
[write it]-COREF
*[∃(A,system(A)):[you],start:[OOV,did]]
[write it]-INTRODUCE
*[write(_,_):[write],∃(A,system(A)):[you],start:[OOV,did]]
[it]-DROP
*[∃(A,system(A),write(_,_)):[you],start:[OOV,did]]
[it]-COREF
*[∃(A,system(A),write(A,_)):[you],start:[OOV,did]]
[]-INTRODUCE
*[∃(_,nonhuman(_)):[it],∃(A,system(A),write(A,_)):[you],start:[OOV,did]]
[]-COREF
*[∃(B,nonhuman(B)):[it],∃(A,system(A),write(A,_)):[you],start:[OOV,did]]
[]-SCOPE
*[∃(A,∃(B,nonhuman(B),system(A),write(A,_))):[you],start:[OOV,did]]
[]-COREF
*[∃(A,∃(B,nonhuman(B),system(A),write(A,B))):[you],start:[OOV,did]]

Discourse Representation Structure Discourse Representation Theory [Kamp and Reyle (1993)] provides a well formalized framework for handling discourse phenomena such as pronoun and presupposition resolutions. Moreover, DRT means of representing meaning, DRSs, can be translated directly into FOL formula. DRSs can be assimilated to boxes having two regions: the top half region contains the *discourse referents* and the bottom half the *conditions*.

4. Training

In training, a training parser is used to generate FOL statistics from DRS annotated training examples.

Spoken Examples in DRS Training format is: **training([phrase], DRS).**
Had we trained the system on recognized output, we could have the following entry:
tr([OOV,did,you,write,it],drs([A,B],[nonhuman(B),system(A),write(A,B)]))

Training Parser While training, a *Training Parser* is used. It tries any possible actions to get to the final parse, without considering any information (such as *statistics*) that could be helpful to guide the parsing process. In training, a *training beam* can be specified. This means that only a certain number of parses will be recorded in the *FOL statistics* for each training example.

FOL statistics The training parser parses the examples to generate the *FOL statistics*. Every step needed to go from the *phrase* to the *DRS* is recorded, as well as final states themselves. Final states are simply the states of the parse stack themselves at the end of the parse. Each of them (actions and final states) are assigned a frequency measure. Each line has either one of the following format (recall that *op* is a container for any action):

```
op(ACTION#PARSE_STACK#INPUT_STRING#FREQUENCY).
final(FINAL_STATE#FREQUENCY).
```

Here are two examples:

```
op(SHIFT(did)#[start:[OOV]]#[did,you,write,it]#0.3).
final([exists(A,exists(B,nonhuman(B),system(A),
 write(A,B))):[you],start:[OOV,did]]#0.2).
```

These statistics are used by the *Stochastic Parser* to compute the best parse.

5. Statistical Parsing

The actual parsing of the input phrase is done by a *Stochastic Parser*. It uses a statistical model to process all the information available from the training phase in order to get the best possible parse (the one with the highest probability). This section describes the statistical parser in some details.

The Search space Like in the training phase, the most obvious way to influence the parse is to tell the parser how many parses it should try before taking a decision. We call it the *search beam* parameter.

Measure of similarity between lists When the parser tries to choose a suitable parse, it must compare list of words (to compare *Actions*, *Parse stacks* or *Input strings*). A good *similarity* measure between lists is essential, but because computing similarity is very demanding on computer resources, one must find a trade-off that preserves computational efficiency. The approach taken is based on n-grams [Cavnar and Trenkle (1994)].

Parametrizing the model The best parse P is found by taking the highest probability P_i among the possible parses (limited by the search beam) available:

$$\mathsf{P} = \max_i P_i \tag{2}$$

Each of these parses P_i have a probability that amounts to combining the probability of the individual *op* or actions together ($\prod_k a_k$, see equ. 4) and adding the probability of the final state ($ProbF$, see equ. 5). These two components are weighted by Pop and $Pfinal$. Those weighting values must be chosen in such a way that translates the importance of the steps needed to get to a final parse compared to the final state itself. In short, the weighting of actions taken together must be high enough to discriminate among similar final states (in terms of probability), should that case arise.
Multiplying by 100 gives a more readable value between 0 and 100.

$$P_i = (Pop * (\prod_k a_k) + Pfinal * ProbF) * 100 \tag{3}$$

The way each *op* a_k is assigned a probability is by taking into account its *similarity* with one of the *ops* in the statistics (P_m) as well as the *frequency* of this *op* ($Frequency$). These two components are also weighted by Pop_sim and Pop_occ. Default values are chosen with respect to how one would want to consider the respective importance of similarity over frequency.

$$a_k = max_m(Pop\_sim * P_m + Pop\_occ * Frequency) \tag{4}$$

Computing the probability of a final parse state is similar to computing the one for actions. A final state probability $ProbF$ is the weighted sum of the most similar final state in the statistical file P_f (see 6) and the frequency of this final state $Frequency$:

$$ProbF = \max_f(Pfinal\_sim * P_f + Pfinal\_occ * Frequency) \tag{5}$$

$$P_f = max_n(sim(t_n, F)) \tag{6}$$

Conventional smoothing techniques are applied whenever necessary.

6. Experimental Results

We have conducted the usual cross-validation testing by dividing our 250 sentence corpus into 10 testing samples of 25 sentences. In training, we produced at most 5 parses for each example. Results for parsing are reported in the following table:

Average/N-Best	1-Best	2-Best	3-Best
Recall-Precision	62%-64%	78%-80%	88%-91%

The parser always produces a valid output, unless there is no keywords introducing a concept in the input: this explains why *Recall* and *Precision* are very similar. The N-Best column is interpreted as follows: the correct output was AMONG the N results produced. A correct output must be exactly like the one produced by the human annotator. The 62% result for 1-Best may seem very modest, but in the context of a conversation, we believe it is more important to focus on the more comfortable 88% 3-Best result. The reason is that in a dialog system, the dialog manager (DM) have access to some sort of history or context to arbitrate between the N-Best semantic interpretations delivered by the semantic parser; in some occasions, it may therefore be preferable for the DM to get more than one parse. We ran the experiment on a 2GHz laptop computer under Sicstus Prolog. Keeping in mind that the system would be run for real-time conversations, we set a threshold of 20 parses or 3 seconds (whatever is reached first) for parsing to complete.

7. Conclusion

In this paper, a new probabilistic framework for semantic parsing is presented. The combination of a *bottom-up* parser and a purely statistical model makes it unique. More precisely, the parser learns efficient ways of parsing new sentences by collecting statistics on the context in which each parsing action takes place. It computes probabilities on the basis of the similarities of those contexts and their frequencies. The result is a simple and robust parser for speech. At this point, we believe that the 1-Best hypothesis recall could be improved by a higher ratio training/lexicon size. However, testing shows excellent results for the 3-best hypothesis. This system offers an approach in which linguistics can play a decisive role. One crucial aspect of the parser, the computation of similarities between context, relies on a good interpretation of linguistic patterns found in phrases, and how those configurations may determine the particular meaning of a word or group of words. This is essential to interpret, and maybe *understand*, conversational speech.

References

Cavnar, W.; Trenkle, J. 1994. N-gram-based text categorization. In: *Proc. of SDAIR-94, 3rd Annual Symposium on Document Analysis and IR*, Las Vegas, US. 161–175

Hermjakob, U.; Mooney, R. 1997. Learning Parse and Translation Decisions From Examples With Rich Context. Technical report: Dept. of Comp. Sciences, Univ. of Texas

Kamp, H.; Reyle, U. 1993. From Discourse to Logic. Dordrecht: Kluwer

Light, M.; Schubert, L. 1994. Knowledge representation for lexical semantics: Is standard first order logic enough?. In: *"Future of the Dictionary" workshop*

Mooney, R.; Tang, L. 2000. Automated construction of database interfaces: Integrating statistical and relational learning for semantic parsing. In: *Proc. of EMNLP/VLC-2000*, Hong Kong. 133–141

Young, S. 2002. Talking to machines (statistically speaking). In: *Proc. ICSLP*. 9–16

MICHEL GÉNÉREUX is a Research Fellow, Information Technology Research Institute, Brighton. He received his Dr.Phil. in Computational Linguistics at the University of Vienna, dealing with Spoken Dialogue Systems. His current research interests concern style in Speech Generation.
E-mail: michel.genereux@itri.brighton.ac.uk

INFORMATION EXTRACTION FROM BIOMEDICAL TEXT: THE BIOTEXT PROJECT

Filip Ginter, Tapio Pahikkala,Sampo Pyysalo, Evgeni Tsivtsivadze
Jorma Boberg, Jouni Järvinen, Aleksandr Mylläri and Tapio Salakoski
Turku Centre for Computer Science (TUCS) and Dept. of IT, University of Turku

Abstract

We study information extraction for identifying protein-protein interactions stated in biomedical text. In this paper, we present an architecture for an information extraction system and discuss our improvements and results pertaining to several components of the system, including information retrieval, named entity recognition, syntactic analysis, and domain analysis. The individual results are discussed in the context of the whole system, and domain adaptations and differences from classical approaches are considered. We combine structural natural language processing with machine learning methods to address the general and domain-specific challenges of information extraction targeting protein-protein interactions.

Keywords: biomedical literature mining, information retrieval, named entity recognition, word sense disambiguation, parsing, parse ranking

1. Introduction

The amount of published knowledge in the biomedical domain is overwhelming and grows at an unprecedented rate. Although many databases collecting biomedical knowledge exist, their coverage is limited and manual identification of e.g. protein-protein interactions requires significant human effort. Freeform text remains a main source of information and thus Natural Language Processing (NLP) and Information Extraction (IE) methods are required to facilitate automated processing and structured access to the knowledge. The BioText project aims at developing NLP methods and resources for biomedical text mining as well as adapting existing methods to take into account the specific properties of the biomedical text domain. This paper gives an overview of our approach, the developed methods and the key results of the project.

Our overall goal is the development of a modular system that processes biomedical text, such as abstracts contained in the PubMed literature database, and extracts the protein-protein interactions stated therein. The system consists of the following major subsystems: Information Retrieval (IR), Named Entity (NE) recognition, syntactic analysis, and pattern-based domain analysis. We apply machine learning approaches such as Bayesian classification, Support Vector Machines (SVM) (see e.g. Vapnik 1998) and Regularized Least-Squares (RLS) (see e.g. Poggio and Smale 2003) as well as

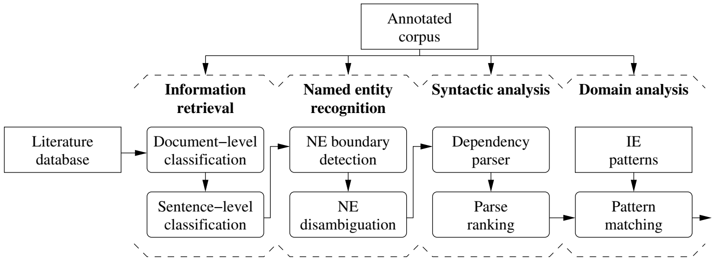


Figure 1: The IE system architecture

structural linguistic methods such as dependency-based syntactic analysis. We also develop methods that combine the two general approaches, taking advantage of both explicit linguistic knowledge and machine learning. For a recent thorough review of related work in Bio-NLP, see for example Cohen and Hunter (2004).

The architecture of our IE system is illustrated in Figure 1. The following sections describe the components of the system in detail.

2. Annotated domain-language corpus

An annotated domain-language corpus is necessary to facilitate the development and evaluation of the various parts of IE systems. We have created a corpus of biomedical English focused on protein-protein interactions. The corpus consists of 1100 sentences manually annotated at three levels: NEs, dependency syntax, and entity interactions. It can thus provide data for the development and evaluation of all of the key components of the IE system. Further, as all the levels of annotation are provided for a single set of sentences, the corpus allows the components of the IE system to be tested not only individually but also as an integrated whole. The corpus is described in detail in Ginter et al. (2004d) and will be made publicly available at `http://www.cs.utu.fi/bdb`.

3. Information retrieval

Many of the steps in the IE system, e.g. the full syntactic analysis, are computationally costly. Fully processing a large literature database such as PubMed, which contains 7.5 million article abstracts, is thus not practical. We therefore study IR methods that retrieve from publications only the sentences which are relevant to the domain of interest. While many standard approaches to IR have been described in literature, we study methods that utilize information specific to the biomedical domain. In Ginter et al. (2004c), we introduced a method applicable to the classification of PubMed-indexed articles. We devised a scheme for transforming the MeSH biomedical ontology used to index PubMed articles, and showed that the ontology transformations lead to an increase in classification performance. To identify individual sentences likely to discuss protein-protein interactions, we also introduced a method in which known protein names, verbs specific to protein-protein interactions, and their mutual positions in the sentence are used as features for a rough-set based classifier (Ginter et al. 2004b).

4. Named entity recognition and disambiguation

NE recognition can be divided into two subtasks: determining the boundaries of the NEs and classifying the entities into classes such as genes and proteins. Both problems can be addressed using Word Sense Disambiguation (WSD) methods. Much of the ambiguity in biomedical text is caused by inconsistent or non-existent naming conventions. Further, capitalization and other surface clues are not reliable indicators of entities in the domain. For example, there exist Drosophila gene names such as *white* and *cycle* which can be confused with the ordinary meanings of these words. We use machine learning methods with particular focus on kernel-based learning algorithms (see e.g. Schölkopf and Smola 2002) to address the problem of WSD.

In Ginter et al. (2004a), we introduced a statistical classification method and a weighted bag-of-words representation, where the context words are weighted so that the words located closer to the ambiguous word receive higher weights. The new method was shown to improve the classification performance in gene/protein name disambiguation from 79% to 82% accuracy.

We have adapted the weighted bag-of-words approach for SVM classifiers and applied them to the problem of gene/protein name disambiguation, improving the performance, measured as the area under the ROC curve (AUC), from 80% to 85% (Pahikkala et al. 2004). We have also introduced a position-sensitive kernel function which generalizes over the ordinary bag-of-words, position-sensitive bag-of-words and weighted bag-of-words approaches (Pahikkala et al. 2005b). Considering context-sensitive spelling error correction as a WSD problem, it was demonstrated that the position-sensitive kernel improves the performance of the SVM classifier from 94% to 98% (AUC). The results reflect the difficulty of the biomedical disambiguation tasks as well as demonstrate the applicability of the method to other domains.

In Pahikkala et al. (2005a), we further analyze this kernel function and construct smoothed word position-sensitive as well as smoothed word position- and distance-sensitive representations of our training data using kernel density estimation techniques (see e.g. Silverman 1986). For the Naïve Bayes classifier, these representations were used to obtain class-conditional probabilities of word-position features. We demonstrate with the Senseval-3 data that the kernel improves the classification performance of SVMs compared to the ordinary Bag-of-Words kernel and furthermore improves the classification performance of the Bayes classifier given the kernel-smoothed data representation.

5. Syntactic analysis

In this section, we present our choice of parser and the architecture of the syntactic analysis component. We also illustrate our use of machine-learning methods to improve the performance of a parser based on a hand-written grammar.

5.1. Parser

Our analyses suggest that general English parsers may not be well applicable to biomedical English, and that adaption to the domain is required (Pyysalo et al. 2004: 2005). Moreover, a statistical inference of the domain grammar is infeasible as the amount of treebank data in the domain is very limited—the largest domain treebank is the GENIA treebank[1] with 1700 sentences. This motivates the choice of a parser based on a hand-written grammar that can be manually adapted to the domain; in the BioText project, we have decided to use the Link Grammar (LG) parser of Sleator and Temperley (1991). The LG parser is a full dependency parser with broad coverage of newswire English. LG has recently received significant attention in the Bio-NLP domain, see for example Alphonse et al. (2004).

The architecture of our syntactic analysis component built around LG is as follows. First, the input sentences are tokenized in a separate tokenization step: The tokenization model originally used by LG was found unsuitable for many common features of biomedical text and replaced with an external tokenization system. After tokenization, we have chosen to augment the parsing system with separate preprocessing and postprocessing stages. In preprocessing, input sentences are simplified by replacing detected NEs with single tokens recognized by the parser, as well as by removing citations and other features for which the parser has no support and which can be naturally captured using regular expressions. Postprocessing is applied after parsing to restore the original sentence text. To improve the applicability of LG to the biomedical domain, we have implemented a number of the modifications proposed in Pyysalo et al. (2004). While the work on parser adaptation is still undergoing, preliminary evaluation suggests that the implemented modifications increase the fraction of recovered correct dependencies from 73% to 78% in the parse ranked first by the built-in heuristics of the LG parser. The parser generates all the alternative parses allowed by the grammar; the respective improvement for the best generated parse is from 82% to 89%.

The domain analysis is performed on the first parse returned by the syntactic analysis component. We have found that the heuristic parse ranking of LG often performs poorly, failing to rank the best parses first. To address this issue, we are developing a machine-learning approach for parse ranking that is applied after post-processing.

5.2. Parse ranking

The task of recognizing the best parses among a set of alternative parses for a single sentence can be cast as a ranking problem. We are currently developing a ranking machine based on the RLS algorithm. Our methodology couples RLS, different ranking performance measures and grammatically motivated features. To convey the most important information about parse structure to the ranking machine, we apply features such as grammatical bigrams, link types (the grammatical roles assigned to the links), a combination of link length and link type, part-of-speech information, and several additional attributes. Each parse is assigned a penalty based on the number of incorrect links. We are also studying a scoring approach where additional information about link types is used in penalization.

The developed method, Regularized Least-Squares Ranking (RLSRa), is a special case of ordinal regression where performance evaluation is based on the rank correlation measure of Kendall (1970), scaled between zero and one. Preliminary evaluation of RLSRa against LG parser built-in heuristics indicates a performance improvement from 55% to 70% using our method. Furthermore, RLSRa provides a reliable ranking solution in application to sparse biomedical datasets.

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/GTB.html

6. Domain analysis

To extract factual knowledge from the parsed sentences, we are developing a set of hand-written patterns. Each pattern specifies a substructure of the linkage (the graph that represents an LG dependency parse) that is likely to state a protein-protein interaction. A successful match of a pattern in a linkage corresponds to an identified interaction. We chose to create high-precision patterns to minimize the number of false positive matches. High precision typically implies low recall; however, due to the large amount of published literature, the system can be given more than one opportunity to extract most of the interactions as they are likely to be stated in several publications. Processing more data can thus diminish the low recall problem to some extent.

The choice of parser has an obvious influence on the nature of the patterns and the formalism in which the patterns are expressed. Since we chose a full dependency parser, it is natural to represent both the linkage and the patterns in terms of relations on the set of words and link types. This representation is naturally and straightforwardly expressed in a declarative language such as Prolog. Each linkage and each pattern are thus described as a set of predicates, and the unification mechanism of Prolog provides the pattern matching mechanism.

7. Conclusions and future work

We have described our work in biomedical IE, presented the architecture of an IE system targeting protein-protein interactions, and discussed each of its components. For each part of the system, we have presented our approach, summarizing improvements and key results. Currently, we are focusing on finishing the domain adaptation of the syntactic analysis component and the development of IE patterns. The implementation of an integrated system that combines the discussed components remains future work.

Acknowledgements

This work has been supported by Tekes, the Finnish National Technology Agency.

References

Alphonse, Erick; Aubin, Sophie; Bessiéres, Philippe; Bisson, Gilles; Hamon, Thierry; Lagarrigue, Sandrine; Nazarenko, Adeline; Manine, Alaine-Pierre; Nédellec, Claire; Vetah, Mohamed Ould Abdel; Poibeau, Thierry; Weissenbacher, Davy 2004. Event-Based Information Extraction for the biomedical domain: the Caderige project. In: *Proceedings of the JNLPBA workshop at COLING'04, Geneva*. 43–49

Cohen, K. Bretonnel; Hunter, Lawrence 2004. In: Dubitzky, Werner; Pereira, F. (eds.), *Artificial intelligence and systems biology*, Kluwer Academic Publishers

Ginter, Filip; Boberg, Jorma; Järvinen, Jouni; Salakoski, Tapio 2004a. New techniques for disambiguation in natural language and their application to biological text. In: *Journal of Machine Learning Research* **5**, 605–621

Ginter, Filip; Pahikkala, Tapio; Pyysalo, Sampo; Boberg, Jorma; Järvinen, Jouni; Salakoski, Tapio 2004b. Extracting protein-protein interaction sentences by applying rough set data analysis. In: *Proceedings of RSCTC'04*: Vol. 3066 of *Lecture Notes in Artificial Intelligence*, Springer, Heidelberg. 780–785

Ginter, Filip; Pyysalo, Sampo; Boberg, Jorma; Järvinen, Jouni; Salakoski, Tapio 2004c. Ontology-based feature transformations: A data-driven approach. In: *Proceedings of EsTAL'04*: Vol. 3230 of *Lecture Notes in Artificial Intelligence*, Springer, Heidelberg. 279–290

Ginter, Filip; Pyysalo, Sampo; Heimonen, Juho; Boberg, Jorma; Järvinen, Jouni; Salakoski, Tapio 2004d. Bio Dependency Bank: a dependency corpus for information extraction in the biomedical domain. Submitted.

Kendall, Maurice G. 1970. Rank Correlation Methods. Griffin, London

Pahikkala, Tapio; Ginter, Filip; Boberg, Jorma; Järvinen, Jouni; Salakoski, Tapio 2004. Contextual weighting for support vector machines in literature mining: an application to gene versus protein name disambiguation. Submitted.

Pahikkala, Tapio; Pyysalo, Sampo; Boberg, Jorma; Mylläri, Aleksandr; Salakoski, Tapio 2005a. Improving the performance of Bayesian and support vector classifiers in word sense disambiguation using positional information. Submitted.

Pahikkala, Tapio; Pyysalo, Sampo; Ginter, Filip; Boberg, Jorma; Järvinen, Jouni; Salakoski, Tapio 2005b. Kernels incorporating word positional information in natural language disambiguation tasks. In: *Proceedings of FLAIRS'05*, Clearwater Beach, Florida. To appear.

Poggio, T.; Smale, S. 2003. The mathematics of learning: Dealing with data. In: *Amer. Math. Soc. Notice* **50(5)**, 537–544

Pyysalo, Sampo; Ginter, Filip; Pahikkala, Tapio; Boberg, Jorma; Järvinen, Jouni; Salakoski, Tapio 2005. Analysis of two dependency parsers on biomedical corpus targeted at protein-protein interactions. Submitted.

Pyysalo, Sampo; Ginter, Filip; Pahikkala, Tapio; Boberg, Jorma; Järvinen, Jouni; Salakoski, Tapio; Koivula, Jeppe 2004. Analysis of link grammar on biomedical dependency corpus targeted at protein-protein interactions. In: *Proceedings of the JNLPBA workshop at COLING'04, Geneva*. 15–21

Schölkopf, Bernhard; Smola, Alexander J. 2002. Learning with kernels. MIT Press, Cambridge

Silverman, B. W. 1986. Density estimation for statistics and data analysis. Chapman & Hall, London

Sleator, Daniel D.; Temperley, Davy 1991. Parsing English with a link grammar. Technical Report CMU-CS-91-196: Department of Computer Science, Carnegie Mellon University, Pittsburgh

Vapnik, Vladimir 1998. Statistical Learning Theory. Wiley, New York

F. GINTER, T. PAHIKKALA, S. PYYSALO, AND E. TSIVTSIVADZE are postgraduate students at Turku Centre for Computer Science (TUCS), holding MSc. degrees in computer science and working full-time for the BioText project since 2001, 2002, 2003, and 2004, respectively. E-mail: firstname.lastname@it.utu.fi.

J. BOBERG, J. JÄRVINEN, A. MYLLÄRI, AND T. SALAKOSKI are lecturers at the Dept. of IT, University of Turku, holding PhD. degrees in computers science, mathematics, mathematics, and computer science, respectively. T. Salakoski is further a professor of computer science and vice-head of the department. E-mail: firstname.lastname@it.utu.fi.

WORD MODEL-DETERMINED SEGMENTAL DURATION IN FINNISH SPEECH SYNTHESIS AND ITS EFFECT ON NATURALNESS

Jussi Hakokari[1], **Tuomo Saarni**[2], **Mikko Jalonen**[2], **Olli Aaltonen**[1], **Jouni Isoaho**[2], **Tapio Salakoski**[2]

[1]Phonetics Laboratory, Department of Finnish and General Linguistics, University of Turku ([Turku]Finland)

[2]Department of Information Technology, University of Turku ([Turku]Finland)

Abstract

The 50-year-old concept of formant synthesis was under much scientific scrutiny until it became apparent that naturalness was hard to attain using rule-based synthesis methods. Formant synthesis was essentially an intelligible approximation, rather than imitation, of human speech. Our current research is a revisitation to the rule-based formant synthesis. At the moment, due to advances in computer technology, we are in a better disposition to develop naturalness in rule-based text-to-speech systems. While research teams in various countries have made similar efforts, our project is unique in Finland and, more importantly, as regarding the Finnish language. The duration of individual phones is important to naturalness in Finnish. We have extracted data from an extensive single-speaker Finnish speech corpus and created word models to prescribe each phone a duration depending on its position within a word. In this paper, we will describe our approach, present the preliminary results of listening tests and discuss the potential of word models in improving naturalness in text-to-speech systems.

Keywords: rule-based, formant, text-to-speech, consonant/vowel, pattern, Klatt, TTS

1. Introduction

The Finnish language exhibits contrast between phonemically short and long segments (also called chronemic contrast). This contrast applies to all vowels and most consonants. The short vowels are generally more central in vowel space while the long ones are peripheral (Wiik 1965, Lennes 2003). The decisive factor, however, is duration (Wiik 1965) and Finnish speakers are unaware of any qualitative differences between the two chronemic variants. The acoustic difference between short and long phonemes is not linear, but relative to the segment's position in the syllabic structure of the word, and to some degree, the word's position within a sentence. The aim of the listening test in this study is to investigate whether or not varying segmental duration is useful in improving naturalness and rhythm in a TTS (text-to-speech) application. Our long term objective is to prepare a TTS system that will not only introduce varying mean durations, but also quantitative and qualitative reduction characteristic of natural speech as well as sentence context sensitive modeling of duration and fundamental frequency.

2. Methods

2.1. Data analysis

We have examined data on segmental durations presented in Lehtonen (1970), and datamined a single speaker speech corpus of 692 segmented and annotated sentences prepared and studied by Vainio (2001). The corpus contains approximately 6500 words, and is read aloud by a 39-year-old male, a native Finnish speaker from Helsinki. Sentence lengths in the corpus vary from 2.18 s to 20.00 s, and the database adds up to approximately 69 minutes of recording.

All the consonant/vowel patterns of individual words were extracted from the speech corpus automatically using software designed specifically for the task. The software makes use of the original annotation provided with the corpus, and allocates each consonant/vowel pattern found indiscriminately into its own class, maintaining duration information of each phone. A consonant/vowel pattern, together with the mean durations of each segment is referred to as 'word model'. For instance, our data contained 125 occurrences of VCCV – pattern words such as <usko> (*faith*) and <akka> (*an old woman*). Any word with a single short vowel, a geminate or two consecutive consonants, and finally another single short vowel will fall into this category. Our data added up to a mean duration structure of 78 ms for the first vowel, 61 ms for the first consonant, 66 ms for the second consonant (or a total of 127 ms for a geminate), and 48 ms for the final vowel. We have been able to establish ~1100 different word models.

To implement, we have prepared a synthesizer that automatically determines each segment's duration by matching the word against its corresponding model in the database. For instance, in CVCCV words, such as <miksi> (*why*), the first vowel has a mean duration of 73 ms, whereas the second has a mean of only 53 ms (208 tokens). The synthesizer produces the closest match possible to the values in the database; the duration of an individual segment varies to some degree due to F0-induced differences in wave length. We are unaware of any other Finnish TTS taking that kind of within-word environment into consideration.

2.2. Stimulus generation

The first set of stimuli for the listening test was produced using the original, unaltered configuration of the synthesizer which produces speech signal with fixed segmental duration. There is a cascading F0 contour (100-120-80 Hz). The second set of of stimuli was generated with an improved model, that introduces more variation in F0 (100-140-80 Hz) and word modeling to determine segmental durations. The original configuration produces greater segmental and overall durations (30–35 % longer) than the improved one; the first set of stimuli was adjusted to the same length with the second using a PSOLA (Pitch-Synchronous Overlap and Add) algorithm. The operation maintains the spectral characteristics and fundamental frequency of the signal. Additionally, we have had to make minor adjustments to how transitions between phones are realized because the word models cannot be implemented to original system as such. The 16 stimuli represented four categories: four single words (0.74 s – 1.03 s in duration), four short sentences containing short words (1.11 s – 2.45 s), four longer sentences with long words (3.64 s – 4.00 s), and four sentences of medium length with no particular constraints (2.38 s – 2.71 s). All of the sentences were in Standard Finnish and adapted from newspaper articles; some of the sentences represented informal literary style while the majority were formal.

All the synthetic stimuli were produced with the data driven formant synthesis program under development at the University of Turku. The program uses SenSyn 1.1

software for signal generation from a parameter file. SenSyn 1.1, by Sensimetrics Corporation, is based on KLSYN88 synthesizer (Klatt 1982). The resulting signal has a sample rate of 10 kHz. The organization structure of the overall system is illustrated in figure 1.

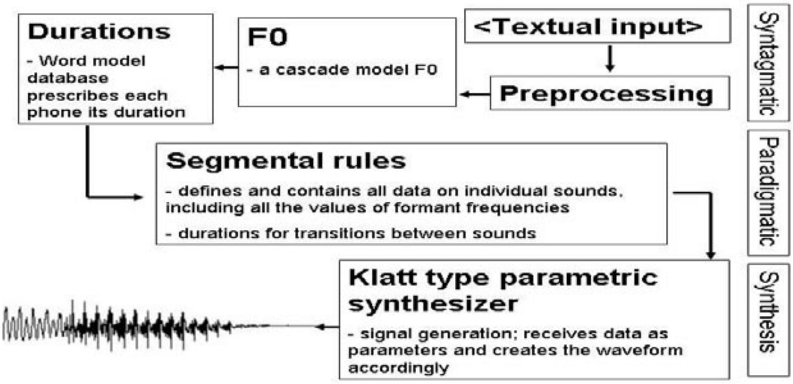

Durations
- Word model database prescribes each phone its duration
F0
- a cascade model F0
<Textual input>
Preprocessing
Segmental rules
- defines and contains all data on individual sounds, including all the values of formant frequencies
- durations for transitions between sounds
Klatt type parametric synthesizer
- signal generation; receives data as parameters and creates the waveform accordingly
Syntagmatic
Paradigmatic
Synthesis

Figure 1. A modular representation of the TTS system

2.3. Participants

None of the participants had been exposed to the synthesis in question previously, and they were neither specialists in language nor synthetic speech. They were asked about their primary and secondary dialect background, since prosody, segmental duration included, is dialect sensitive in Finnish. There were 10 women from ages 20 to 54. Two of the participants were left-handed, and no one reported any deficit in hearing or language.

2.4. Listening test procedure

Due to similarity and brevity of the stimuli and naïve participants we opted for a forced choice paradigm instead of Category Estimation as the evaluation method. The participants heard two words or sentences of identical length successively. Their task was to identify which one of the two sounded more natural. They had transcripts of the sentences to prevent intelligibility issues from diverting them from their task. They were specifically instructed to judge how well the stimuli corresponded to human speech patterns, instead of how clear, pleasant, or intelligible they were. The participants were presented the stimuli in a pseudorandomized order, so that the original and improved versions would not occur consecutively. Presentation order was the same for all participants. The participants judged a total of 16 stimulus pairs.

The session lasted for 15 minutes, and took place in an ordinary laboratory room with no external distraction or noise. The uncompressed sound files (.wav) were played with an ordinary laptop computer and Labtec LCS-1060 loudspeakers. Volume was adjusted to be as loud as possible without causing distortion in the signal or discomfort in the participants.

3. Results and discussion

By data analysis, we have found that even in perfectly intelligible sentences read clearly aloud there is considerable overlap between short and long phonemes. In other words, a short phoneme may be longer in duration in one position than a long phoneme in another, and that alone does not cause intelligibility issues. For instance, the short vowel /o/ varies from a (reduced) single periodic waveform to 200 ms, while the long vowel /o:/ varies from 54 ms to 294 ms. The overlap applies to all the phonemes in the corpus which exhibit chronemic contrast. Since perception of natural speech is adapted to relative segmental duration, instead of relying on absolute duration, we expected word models to affect naturalness in synthetic speech as well.

The results of the listening test, presented in figure 2, were generally ambivalent towards the use of word models. Only 66 of the 160 responses (41,25 %) preferred the improved configuration. 76 (47,50 %) of the responses preferred the stimulus presented first, and 84 (52,50 %) preferred the second; there was no bias concerning order of presentation. 4 out of the 16 improved stimuli were judged better than the original ones, two of them unanimously. 3 were at chance level, and 9 were deemed worse.

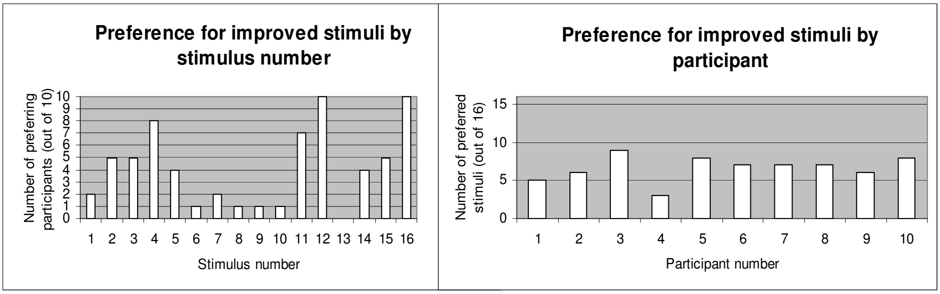


Figure 2. Numerical data

The five least favoured improved stimuli (zero or one preferring responses) were either in the single word or the short sentences with short words category. The two improved stimuli preferred unanimously were of the long sentences with long words category. The other two rated above chance level (eight and seven preferring responses) were of the unconstrained and the short sentences categories. The effect of dialect background is hard to determine due to the homogeneity of the participants. However, the improved set of stimuli was clearly rated lowest by the Southeastern Finnish (South Karelian) speaker, the only non-Southwesterner in the group, who preferred only 3 out of 16 improved stimuli. The following examples, from 1 to 4, are the sentences that were judged more natural in the improved version; the English translations are in italics. Examples from 5 to 9 are the sentences that were judged unnatural the most.

(1) Radion faktasarjan palkinnon sai kulttuuriohjelmien dokumentti. *A documentary by cultural programs (a department of the Finnish Broadcasting Company) won the factual series of the radio competition.*

(2) Hannover on Euroopan tärkeimpiä kansainvälisiä keskuksia. *Hanover is one of the most important international hubs in Europe.*

(3) USA:n (<uuesaan>) joukot toimivat ilman YK:n (<yykoon>) lupaa. *The US forces operate without UN approval.*

(4) Ei liian erikoinen eikä liian tavallinen. *Not too special or ordinary.*

(5) Sunnuntaisin. *On Sundays.*

(6) Peruskorjaus. *Renovation.*

(7) Osakkeenomistaja. *Shareholder.*

(8) Hän ei ole enää olemassa. *(S)he exists no more.*

(9) Miksi Turku ei kasva? *Why isn't Turku getting bigger?*

In the light of present data, word modeling does not appear to improve naturalness universally; the results show responses below chance level. However, we can see improvement in the category of long sentences with long words. We can identify several possible causes for the conflicting results.

At this point it is unclear how the word models will affect once other naturalness features, prosodic and segmental, are implemented. The current system incorporates word models into a synthesizer that is designed to handle fixed segmental durations. Less peripheral formant values typical of ordinary speech (Lennes 2003), a better modeling of F0 contours, and somewhat longer segmental durations might suit the word modeled synthesis better. At the moment the synthesis uses highly peripheral (great acoustic distance between speech sounds) formant values to promote intelligibility.

The more common models, essentially short words, are based on mean values calculated from a sample size up to 287 tokens (the model CV). The longer and strongly inflected word forms, the word models of which are based on only one token (an occurrence of the word in the corpus), make up ~59 % of the database. Those word models carry greater within variation in segmental duration, while within variation has been neutralized due to averaging in the others. That may explain why longer words scored better in the preliminary experiment. In future experiments, it would be worth the while to base all word models on single tokens. The current database is primitive in that it treats all consonants and all vowels equally. In addition, there is no distinction of sentence environment; words occurring in the beginning and the end of sentences are all included into the database without any special tagging. A more detailed datamining could produce contextual classes for the word model database that would differentiate, for instance, voiceless stops from fricatives.

Stimuli used appeared to have too fast an articulation rate (up to 407 syllables per minute). The participants reported they had difficulties in judging the stimuli. They may have found fast, synthetic speech they are unaccustomed to confusing, and picked up a strategy that favors one set of stimuli over the other by some factor other than rhythm and naturalness of speech. The second set of stimuli could be lengthened to match the overall duration of the first in a future study, instead of shortening the first set. This would make the task of judging synthetic speech easier to the naïve participant. Another line of study could make use of prolonged exposure. First, the participants could judge entire paragraphs of text or newspaper articles instead of single words and sentences. Second, the participants could become familiarized to synthetic speech beforehand to avert confusion. With synthetic speech, certain monotonous or reoccurring elements may become irritating after a while. Word models may help to create a less predictable and monotonous synthesis.

Segmental durations are subject to considerable dialectal variation in Finnish. Even if the word models do not contribute a great deal to speech quality or naturalness per se, it is worth further attention to investigate whether they can be used to emulate speakers from different regions or localities. It may be of consequence from the vantage point of Finnish TTS product development, as some end users may prefer to have the synthesizer speak in a manner familiar to them. If dialectal variation, individual speakers or styles can be imitated by the method, it is, from a scientific point of view, a discovery itself. Nine out of ten participants in the present study reported to speak one of the Southwestern dialects (the remaining one a Southeastern speaker), and none were speakers of Helsinki dialect the word models are based on.

4. Summary

In this paper, we have studied the use of word models to prescribe segmental duration in Finnish language speech synthesis. First, we established an initial set of word models by datamining a single speaker speech corpus. Second, we have implemented the word models into our developing rule-based TTS system. Third, we have done a preliminary listening test to examine their effect on naturalness in synthetic speech.

Our studies show there is indisputable overlap between long and short phonemes. Chronemic contrast has been handled in speech synthesis by giving short and long speech sounds fixed durations. Naturalness may be improved by additional modeling of segmental duration according to how it is realized in natural speech. The listening test, however, showed that the method does not give a straightforward advantage. Instead, the results are ambiguous but suggest the word models are most effective when applied to long words in long sentences. The synthesizer in its current, experimental configuration does not improve naturalness in shorter utterances. There is potential in at least a selective implementation of varying segmental duration, and the matter requires a more detailed and more comprehensive investigation. Word modeling may be sensitive to qualities in synthetic speech we have not yet taken into consideration.

References

Klatt, Dennis 1980. Software for a cascade/parallel formant synthesizer. *Journal of the Acoustical Society of America* 67. 971–995

Lehtonen, Jaakko 1970. Aspects of quantity in standard Finnish. Jyväskylä: University of Jyväskylä.

Lennes, Mietta 2003. On the expected variability of vowel quality in Finnish informal dialogue. In: Solé, M., Recasens, D., Romero, J., (eds.) *Proceedings of the 15th International Congress of Phonetic Sciences (ICPhS).* 2985–2988.

Vainio, Martti 2001. Artificial neural networks based prosody models for finnish text-to-speech synthesis. Helsinki: University of Helsinki.

Wiik, Kalevi 1965. Finnish and English Vowels. Turku: University of Turku.

JUSSI HAKOKARI is a research assistant at the Phonetics Lab (Department of Finnish and General Linguistics) at the University of Turku in Turku, Finland. He received his B.A. (phonetics) at the University of Turku, dealing with speech synthesis. His research interests concern speech synthesis and speech acoustics. His master's thesis focuses on Finnish language TTSs and rule-based formant synthesis. As a lecturer, he has taught acoustic analysis, pronunciation and phonetic transcription at the University of Turku. E-mail: jussi.hakokari(à)utu.fi.

TUOMO SAARNI is a research assistant at the Department of Information Technology at the University of Turku in Turku, Finland. He received his B.Sc. (computer science) at the University of Turku, dealing with visibility algorithms in 3D computer graphics. His research interests concern automatic analysis methods in developing speech synthesis. His master's thesis focuses on Finnish language TTSs, rule-based formant synthesis and data mining of natural speech corpora. E-mail: tuomo.saarni(à)utu.fi

WORD SENSE DISAMBIGUATION CORPUS OF ESTONIAN

Kadri Kerner, Kadri Vider
University of Tartu (Estonia)

Abstract

The research group of computational linguistics of the University of Tartu has developed Word Sense Disambiguation Corpus of Estonian (WSDCEst). During four years 100 000 running words are looked over and all content words in texts are manually annotated according to EstWN word senses. The source texts are mostly fiction. Paper gives quantitative analysis of corpus and focuses on some inspiring and linguistically relevant ideas and hypoteses.

There were significant inconsistencies in opinions of these people, who disambiguated the texts. This shows us the most problematic entries in EstWN, the need to reconsider the borders of meaning of some concepts. Part of our research focuses on exploiting agreement and disagreement of human annotators: are there any remarkable and important sense clusters.

Manual sense tagging refers to problems in EstWN like missing examples, overlapping synsets of explanations and over-grained senses. Sense clusters are made by processing the disagreement files and the most frequent words are analyzed by describing lexical relations like autohyponymy and sisters (co-hyponyms).

Keywords: word sense disambiguation, semantic annotation, corpora, Estonian Wordnet, sense clusters

1. General[1]

For several language technology applications, it is important to make sure in which sense each word is meant. Demand for systems able to resolve this problem originated SensEval – international organization devoted to the evaluation of Word Sense Disambiguation Systems (SensEval). Part of Word Sense Disambiguation Corpus Estonian was distributed as Gold Standard as well as Test Corpus for Estonian task in Senseval-2 competition in 2001 (Kahusk et al. 2002).

The problem of semantic disambiguation (tagging and annotation) is tightly connected to morphological and syntactic disambiguation, but is more complicated. It is even argued, that the concept of a word sense is questionable and depends on goals of

[1] This paper is based on work supported in part by the Estonian Science Foundation under grant No 5534 “Concept based resources and processing tools for the Estonian language” and by Estonian State Target Financing R&D project No 0182541s03 “Computational models and language resources for Estonian: theoretical and applicational aspects” and by Governmental programme "Eesti keel ja rahvuslik mälu" sub-project “Language Technology: Semantic analysis of Estonian simple sentence”.

disambiguation (Kilgarriff 1997). However is word sense disambiguation (WSD) task of interest in lexicography and lexical semantics.

2. Sense-tagged corpora

There are two main approaches one can take to the order in which words are tagged in texts (Langone et al. 2004; Kilgarriff 1998). In the sequential approach (also termed 'textual' or 'all-words-task') annotator tries to assign the context-appropriate sense to each open class word as it is encountered. The targeted approach (also termed 'lexical' or 'lexical sample task') involves tagging all corpus instances of a pre-selected word.

Some semantically annotated corpora, e.g. SEMCOR in English, are tagged sequentially and we chose same approach for WSDCEst. Other corpora, for example HECTOR in English, use lexical choice and receive very detailed sense-distinctions of particular word. It is good to know, that SEMCOR as well as HECTOR corpora use English WordNet sense distinctions.

It should be kept in mind, that not all words can be disambiguated, but only content words. Although normally nouns, verbs, adjectives and adverbs are considered as content words (see e.g. Stevenson, Wilks 2001), in WSDCEst only nouns and verbs were subject to disambiguation. We use EstWN as sense distinction resource (see Kahusk et al. 2005 in this volume), but adverbs are not represented in EstWN yet, and there is too little number of adjectives.

3. Multilevel approach to WSD

Word sense disambiguation is closely connected to morphological and syntactic disambiguation. Stevenson and Wilks (2001) propose multilevel approach to WSD. Semantic, sometimes even pragmatic information can be derived from hypernymy hierarchies, and syntactic information can be read from morphological analysis.

3.1. Morphology

Lexical entries (literals) in EstWN are presented nominal singular form for nouns and supine form for verbs. In real texts, the words are mostly in their full richness of forms. Lemmatizing and part-of-speech-tagging are made with Estmorf tagger (Kaalep 1997). In sense annotating we considered only nouns (_S_ com) and non-auxiliary verbs (_V_main or _V_ mod).

The modal verbs are explicitly marked in the output of the morphological disambiguator (_V_ mod). When a verb is marked as such, then the senses that don't correspond to the modal senses could be removed, e.g. the word 'saama' has all together 12 senses in the thesaurus, but only 2 of them ('can' or 'may') correspond to the modal use of the word.

The output of the morphological analyzer often contains valuable information for word sense disambiguation. In some cases the word-form used in the text can uniquely specify the sense of the word, although its lemma is ambiguous, e.g. the word 'palk' can either mean salary or log of a tree, but its genitive form is different in each meaning (either 'palga' or 'palgi'). By using only the lemma we ignore this distinction that can be explicitly present in the text.

3.2. Syntax

At the moment the input text contains no information about its syntactic structure, most importantly the verbal phrases and other multi-word units are not marked as such. Also, the syntactic structure can help to reduce the number of possible senses to choose from. For example the most frequent word 'olema' (be, have) has five more frequent senses. Only one sense is present in complementary clauses; 3 senses appear in existential sentences and one in possessive sentences. Linguistic knowledge about the nature of the sentence can help the disambiguation process of human annotator.

4. Texts for WSD Corpus

We chose 43 texts for word sense disambiguation from Corpus of the Estonian Literary Language (CELL) subcorpus of Estonian fiction from 1980s. Each text file contains about 2500 tokens. Most of the texts that are annotated for word senses, are fiction. Total amount of tokens in texts is around 110,000 (depends on calculating punctuation in or out) at present. About 34,5% of them (see Table 1) are annotated content words, whereas its impossible to disambiguate word senses without context we counted items of other part-of-speeches together.

Table 1. Words and senses in WSD Corpus of Estonian

	Nouns (S com)		Verbs (main and modal)	
	Total	Mean per text	Total	Mean per text
Tokens	21373	497,05	17947	417,37
Lemmas	6536	311,98	1649	177,51
Lexical entries found in EstWN	2585	200,07	1261	160,51
Polysemous words	-	223,65	-	227,07
Senses per annotated lexical entry	-	1,12	-	1,42

5. Manual annotation

Twelve linguists and students of linguistics tagged nouns' and verbs' senses in the texts, each text was disambiguated by two persons. Pre-filtering system added lexeme and number of senses for each annotating word found in EstWN. Annotators marked in brackets the sense number of EstWN which matched best with used sense of a word by their opinion. If the word was missing from the EstWN, "0" was marked as sense number, and if the word was found in EstWN, but missed appropriate sense, "+1" was marked. Example (1) presents a sentence „*Neid kentsakaid mõtteid põimin jälle suvel oma kirjutistesse*" (I'm going to weave these weird thoughts into my writings in summer again.) as it occurs in WSD Corpus of Estonian.

If inconsistencies were met, they were discussed until agreement was achieved. On about 20% of cases the disambiguators had different opinions. This shows us the most problematic entries in EstWN, the need to reconsider the borders of meaning of some concepts. Part of our research (Kerner 2004) focuses on exploiting agreement and disagreement of human annotators: are there any remarkable and important sense clusters.

(1)
```
<s>
Neid
   see+d //_P_ dem pl part //
kentsakaid
   kentsakas+id //_A_ pos pl part //
mõtteid
   mõte+id //_S_ com pl part // mõte(1)#@5
põimin
   põimi+n //_V_ main indic pres ps1 sg ps af // põimima(3)#@3
jälle
   jälle+0 //_D_ //
suvel
   suvi+l //_S_ com sg ad // suvi(1)#@1
oma
   oma+0 //_P_ pos sg gen //
kirjutistesse
   kirjutis+tesse //_S_ com pl ill // kirjutis(1)#@1
.
   . //_Z_ Fst //
</s>
```

6. Sense clusters

Sense clusters are made up by processing the disagreement files. The most frequent words are analyzed by looking over different sense numbers that annotators proposed as in Table 2.

Table 2. Sense clusters of HAKKAMA

Combination of sense numbers	Frequency
2 -- 3	17
2 -- 5	10
2 -- 6	9
3 -- 5	3

There is little disagreement among word senses that doesn't include autohyponymy and/or sisters (co-hyponyms). Also a very important observation is that human annotators disagree less when all the representation fields of EstWN are properly filled (hyperonym (s), synset, definition, explanation). The research referred to the fact that explanations seem to be very important for human annotators. When adding missing explanations, the difference between senses becomes more definite. The fact that some words are not highly polysemous indicates usually (but not always) to a minor disagreement of human annotators.

Manual sense tagging refers to problems in EstWN like missing examples, overlapping synsets or explanations and over-grained senses. In many cases it is impossible to determine the one and only sense. Sometimes it is even not necessary (Vider et al 2003: 316-317) and sometimes the nearby context allows different senses.

It is difficult to distinguish word senses that are detectable in EstWN but not visible in the real usage of text (or language). In some cases the disagreement between human annotators arises, because of the lack of lexicographical knowledge (or the human annotator is somewhat superficial).

Exploiting sense clusters can be helpful in referring to insufficiency of EstWN. For example, if all the sense numbers of a word combine with each other (Table 2), it can be assumed that the distribution of senses is incomplete and needs to be improved. If there is no disagreement among human annotators, then there are no remarkable sense clusters and therefore the senses of this particular word are reasonably distributed (or divided). Also our research showed that words with abstract meanings are difficult to annotate and make up essential sense clusters.

Some researchers (Vider et al. 1998; Vossen et al. 1998:6) claim that words representing so-called Base Concepts are difficult to annotate semantically (apparently because of their broad meanings). This research also confirmed this fact. The boundaries and the area of the usage of a hyperonym or hyponym should be very precisely represented in EstWN. The tendency seems to be that hyponyms as narrower meanings are better to distinguish than hyperonyms. That is the reason, why top concepts tend to combine with many of the different word senses (example in Table 3).

Table 3. Sense clusters of SAAMA

Combination of sense numbers	Frequency
10 -- 11	24
10 -- 9	9
10 -- 2	8
10 -- 6	5
10 -- 3	4
10 -- 7	4

Importance for automatic WSD

Important sense clusters can be effective for improving the processing work of word sense disambiguation system. It is easier for this system to choose the appropriate sense if the word senses are not too over-grained; the speed and accuracy of the system will increase. If there are any remarkable sense clusters, it might be useful for the applications of language technology to join these senses. For example, in a machine translation system – if a human annotator can not distinguish all the senses of a particular word, then maybe the machine translation system also should not.

References

CELL = Corpus of the Estonian Literary Language. Retrieved February 15, 2005, from http://test.cl.ut.ee/korpused/morfkorpus/index.html.en

Kaalep, Heiki-Jaan 1997. An Estonian morphological analyser and the impact of a corpus on its development. *Computers and the Humanities*, 31, 115–133

Kahusk, Neeme; Orav, Heili; Õim, Haldur 2002. Sensiting inflectionality: Estonian task for SENSEVAL-2. In: *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, 25-28

Kahusk, Neeme; Vider, Kadri 2005. TEKsaurus – the Estonian WordNet online. *This volume*.

Kilgarriff, Adam 1998. Gold standard datasets for evaluating Word Sense Disambiguation programs. *Computer Speech and Language*, 12(3), 453–472.

Kilgarriff, Adam 1997. "I don't believe in word senses." In: *Computers and the Humanities*, 31(2), 91–113.

Kerner, Kadri 2004. Sõnatähendused tekstides ja tesauruses ühestajate erimeelsuste põhjal. (English title: Word senses in texts and in thesauri based on human annotators disagreement) B.A. thesis. (Manuscript.) University of Tartu, Dept. of General Lingustics.

Langone, Helen; Haskell, Benjamin R.; Miller, George A. 2004. Annotating WordNet. In: Meyers, A. (ed). *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation.* Boston: Association for Computational Linguistics. 63–69

SensEval = Senseval web page. Retrieved February 28, 2005 from http://www.senseval.org/

Stevenson, Mark; Wilks, Yorick 2001. The interaction of knowledge sources in word sense disambiguation. In: *Computational Linguistics* 27 (3), 321–349.

Vider, Kadri; Orav, Heili 2003. Idee ja rakenduse vahe tesauruse näitel. In: *Eesti Keele Instituudi toimetised 14. Toimiv keel I. Töid rakenduslingvistika alalt.* Tallinn: Eesti Keele Sihtasutus. 313–322.

Vider, Kadri; Orav, Heili 1998. Sõna tasandilt mõiste ruumi. In: *Keel ja Kirjandus* 1. 57–64.

Vossen, Piek; Kunze, Claudia; Wagner, Andreas; Dutoit, Dominique; Pala, Karel; Sevecek, Pavel; Vider, Kadri; Paldre, Leho; Orav, Heili; Õim, Haldur 1998. *Set of Common Base Concepts in EuroWordnet-2.* Amsterdam: Deliverable 2D001, WP3.1, WP 4.1; EuroWordNet, LE4-8328.

KADRI VIDER is a researcher and a PhD student at Department of General Linguistics, University of Tartu. She received her M.A. in 1999 dealing with senses of Estonian verbs in semantic database such as wordnet. Her research interests concern computational lexicology, lexical semantics and word sense disambiguation. Her doctoral study focuses on senses of Estonian verbs and possibilities to distinguish them in texts. She is member of the board of the Estonian Association of Applied Linguistics. e-mail: kadri.vider@ut.ee

KADRI KERNER is a M.A student at Department of General Linguistics, University of Tartu. She received her B.A in 2004 with the graduation thesis: Word Senses in Texts and in Thesauri based on Human Annotators Disagreement. Her M.A thesis deals with sense clusters of Estonian nouns, grammaticalization, Estonian Wordnet and FrameNet. e-mail: kadri.kerner@ut.ee

ENHANCED INFLECTIONAL STEMS AS SEARCH KEYS IN BEST-MATCH IR

Kimmo Kettunen
University of Tampere, Department of Information Studies

Abstract

The customary answer to the morphological variation of key words in information retrieval (IR) has been a simple type of word form normalization called stemming. First stemmers were introduced for this purpose in the late 1960's. During the 1970's and the 1980's a variety of different stemming methods were implemented. 1980's saw also the in-march of lemmatization: morphological programs were able to analyze inflected word forms and return their dictionary base forms after analysis. These programs were later applied in IR. In Kettunen and colleagues (2005) we (re)introduced a slightly different approach, use of inflectional stems in a best-match IR environment to cover morphological variation of Finnish search keys. The method was found to compare well with lemmatization and it was clearly better than stemming for Finnish in the used test collection. In this paper we shall refine our stem generation method by combining it to a well known language technology tool, namely regular expressions. Our results show that the enhanced stem queries do not outperform basic inflectional stems but they are neither considerably worse. They also perform comparatively well against lemmatization and outperform stemming with Snowball.

Keywords: information retrieval, normalisation of keywords, Finnish language, regular expressions

1. Introduction

Inflection is a common phenomenon in natural languages. Its nature and complexity may vary a lot from language to language. Simply put the situation varies from the two nominal cases of English to tens of cases in many languages. Morphological variation of words can also be handled with different kinds of means in different applications of language technology.

The customary answer to the morphological variation of key words in information retrieval (IR) has been a simple type of word form normalization called stemming. First stemmers were introduced for this purpose in the late 1960's. During the 1970's and the 1980's a variety of different stemming methods were implemented (Frakes 1992). 1980's saw also the in-march of lemmatization: morphological programs were able to analyze inflected word forms and return their dictionary base forms after analysis (Sproat 1992). These programs were later applied in IR, e.g. Alkula (2001) tested lemmatization of Finnish full-text collection and queries thoroughly in a Boolean IR environment.

In 1990's and early 2000, however, the stemming approach was still well alive and stemming was used in IR to morphologically more complex languages than English quite successfully (e.g. Hollink et al. 2004, Kraaij 2004). Information retrieval in languages such as Amharic, Arabic, Dutch, German, Slovene, Turkish and even Finnish has been shown to be quite successful when only a quite simple stemmer was used to normalize the morphological variation of the search keys in indexes and queries.

In Kettunen and colleagues (2005) we (re)introduced a slightly different approach, use of inflectional stems in a best-match IR environment to cover morphological variation of Finnish search keys. The method was found to compare well with lemmatization and it was clearly better than stemming for Finnish in the used test collection. However, the stemmer used, Snowball (Porter 2001), seems to perform better in some collections than in others, and thus it may also be a relevant tool for IR of a morphologically complex language (Airio 2005).

In this paper we shall refine our stem generation method by combining it to a well known language technology tool, namely regular expressions (Friedl 1997, Jurafsky & Martin 2002, Roche & Schabes 1997). In our tests inflectional stems of Finnish nouns are enhanced with regular expressions which will contain the possible continuations of the stem, i.e. its inflectional endings or parts of them.

Our basic hypothesis is that enhanced inflectional stems could make performance of the search stems more effective from an IR point of view. In Kettunen and colleagues (2005) we found out, that inflectional stems behaved well from an IR point of view, but they also made running of the queries slower and sometimes also resulted in quite large queries.

2. Data and methods

The test collection, TUTK, contains a full text database of newspaper articles published in three Finnish newspapers in 1988 – 1992. The newspapers are Aamulehti, Keskisuomalainen, and Kauppalehti and the database consists of 53 893 articles. The articles represent different sections of the newspapers, mostly economics (from all sections of Kauppalehti, some 16 000 articles), and foreign and international affairs (Aamulehti, some 25 000 articles) and articles from all sections of Keskisuomalainen (some 13 000 articles). (Sormunen 2000, Kekäläinen 1999).

Articles of the database are fairly short on average. Typical text paragraphs are two or three sentences in length. The whole database contains 709 317 word-form types (number of the word forms in the inflectional index) which occur as 11 752 290 word-form tokens. The index contains all the word form types of the texts, no stop word lists are used to exclude any words from the index. The average (computed) length of the articles in the database is about 218 words (11 752 290/53 893). Mean length of all the word-form types in the data is 13.14 characters. Mean length of all the word-form tokens weighted with the frequency is 8.03 characters. The topic set consists of 30 topics (Sormunen 2000). Topics are long: the mean length of the original topics is 17.4 words. When stop words are omitted, the mean length is 14.63 words per topic.

As the queries based on the topics of TUTK are unrealistically long, we made short versions of the topics. The short versions were based on the conceptual query plans of the topics, and only the major facets of the queries were taken in the short versions (Kekäläinen 1999: 152 – 153). The mean length of our short queries is 2.93 words, which is quite close to the mean length of web queries (Jansen et al. 2000).

2.1. Relevance levels of TUTK

Sormunen (2000: 63) describes the relevance levels of TUTK test collection. A four point scale 0 – 3 is used, from totally off target documents to highly relevant ones. Relevance levels of the collection are combined TREC wise in this study: relevance level 3 of TUTK is called *stringent*, relevance levels 2 and 3 are joined as *normal* and all the three relevance levels, 1 – 3, are joined as *liberal* relevance.

2.2. Regular expressions

Regular expressions have been studied and used much in computational linguistics since the 1990's. General search programs that can handle regular expressions are common in the Unix world, **grep** (**"g**lobal **r**egular **e**xpression **p**rint**"**) and its relatives being the most common (cf. Dougherty 1992, Friedl 1997). Regular expressions are being suggested in this paper as a method for augmenting stems due to the generality of the formalism and its computational efficiency.

Formally regular expressions belong to regular languages. Regular languages are formalisms that can be recognized with regular automata (Partee et al. 1993: 462 -). Thus they have a solid mathematical foundation. For our purposes it is sufficient to handle regular expressions as a notation that can be used for describing certain phenomena computationally efficiently. Linguistic applications of regular languages can be studied e.g. in Roche & Sable (1997) and Karttunen et al. (1997).

For practical reasons a text-book approach taken in, e.g., Friedl (1997) is sufficient. Simply put we can state e.g. following kinds of expressions with regular expressions:

(1) Character class that consists of *a, b* and *c*: *[abc]*
(2) Character class that consists of anything else but English vowels *[^aeiou]*
(3) Disjunction of strings: *("cat"|"dog"|"rabbit")*
(4) At least one Finnish vowel: *[aeiouyäö]+*

2.3. Enhanced inflectional stems

Enhanced inflectional stems in the study are thus of the form *stem* + *regular expression.* In Kettunen et al. (2005) we produced inflectional stems for query nouns with a stem generator automatically and the whole query process was automated. In this study we produced the final queries for the tests partly automatically, partly manually from the topics of TUTK collection. Parts of the procedures in Kettunen et al.(2005) were re-used to produce first the InQuery query structures with inflectional stems for the query nouns. After this regular expressions were added to inflectional noun stems in queries manually. We had two sets of enhanced stems in the test runs. In *Setting One*, only one character of each possible case ending after the stem was added to the stem using the character set structure *[]*. In *Setting Two,* whole case endings were added to the stems as regular expressions. Restrictiveness of the matching was parametrized with regular expression operators * and + (optionality and obligatority) or *$* (end of word). Verbs were given in their base forms, nominal forms of the verbs which inflect like nouns, were given the same treatment as nouns. As a result we get search keys like in Table 1.

Table 1. Examples of enhanced stems with InQuery's #SYN operator

Shortly enhanced ster s	Fully enhanced stems

#syn(
päätös$
päätös[tkh]*
päätöksi[ieälknst]*
päätökse[elknst]*)
	#syn(
	päätös$
	päätös(tä$\|kin$|hän$)
	päätöksi(in$\|en$|ä$\|l[lt][äe]$|ksi$\|n$|s[st]ä$\|ttä$)
	päätökse(en$\|l[lt][äe]$|ksi$\|n[ä]*$|s[st]ä$\|ttä$))

Here morphological variation of one noun (*päätös,* 'a decision') is accounted for. For all the nouns the base form, nominative singular, was given first with no other endings. On the left side all the varying stems are given with their possible case ending beginnings stated as one character regular expressions using the character class *[]* construction.

For both shortly enhanced stems and fully enhanced stems we have 2 different types of query files: RegStemma+ and RegStemma* are the short forms with either restrictive operator (+) or non-restrictive operator (*). FullRegStemma+ and FullRegStemma* are the respective fully enhanced versions, where restrictiveness of the match is due to the word final operator $ (or its absence).

3. Results

In Kettunen and colleagues (2005) we found out that lemmatization and inflectional stem generation were clearly the two best methods in the TUTK collection over all relevance levels. Stemming with the Finnish stemmer of Snowball was clearly below both methods on all relevance levels tested. Airio (2005), however, tested Snowball with another Finnish collection and got almost as good results with stemming as with lemmatization in the collection. Thus the performance of Snowball might be collection dependent, and it may also be a good method.

In this paper we present new results of enhanced inflectional stems and compare these especially to the two best earlier methods, lemmatization and inflectional stem generation as such. Results of Snowball and plain unprocessed query words are shown merely for base level comparison.

Table 2 presents the P-R results of lemmatization, five stem generation methods, stemming and plain words on liberal, normal and stringent relevance levels.

Table 2. FINTWOL**,** stem generation methods, Snowball and plain words compared on three relevance levels with short queries

	Lib ral relevance Avg precision over reca l levels (%)	**Nor nal relevance** Avg precision over reca l levels (%)	**Strin ent relevance** Avg. recision over recall evels (%)
FINTWOL	30.5	26.4	17.6
Stemma	30.6 (+0.1)	26.9 (+0.5)	17.2 (-0.4)
RegStemma*	28.8 (-1.7)	25.5 (-0.9)	16.9 (-0.7)
RegStemma+	28.0 (-2.5)	25.1 (-1.3)	17.5 (-0.1)

FullRegStemma*	27.8 (-2.7)	24.7 (-1.7)	16.9 (-0.7)
Full RegStemma+	26.6 (-3.9)	23.6 (-2.8)	15.9 (-1.7)
Snowball	23.8 (-6.7)	21.5 (-4.9)	14.3 (-3.3)
PlainInfl	16.1 (-14.4)	15.3 (-11.1)	11.7 (-5.9)
PlainBase	10.7 (-19.8)	9.5 (-16.9)	5.6 (-12.0)

From the Table 2 it can be seen that inflectional stems with short regular expressions perform almost at the same level as lemmatization and normal inflectional stems. Differences are partly quite small, the average precision of RegStemma* on liberal relevance level is 1.7 % units below lemmatization and 1.8 % units below inflectional stems. On normal relevance level the differences are smaller: RegStemma* is 0.9 % units below FINTWOL and 1.4 % units below Stemma. On stringent relevance level RegStemma* is 0.7 % units below FINTWOL and 0.3 % units below Stemma. RegStemma+ was otherwise below RegStemma*, but on the stringent relevance level it outperformed both RegStemma* and Stemma and was only 0.1 % units below FINTWOL.

FullRegStemma+ and FullRegStemma* perform clearly worse than FINTWOL and Stemma on all relevance levels on average. On the stringent relevance the differences are smallest. On liberal and normal relevance levels differences between the performance of FullRegStemma+ and FINTWOL are at greatest.

With short queries the Snowball stemmer performs clearly worse than all the inflectional stem methods. Differences between Snowball and the worst performance of stem methods (= FullRegStemma+), however, are not very great, 1.6 – 2.8 % units. Snowball also outperforms clearly Plain words.

From Table 2 it can also be seen clearly that plain unprocessed query words are not a good method with short queries for a highly inflected language. The query words of PlainInfl were given in the forms they occur in the topic files – partly inflected and partly base forms – and this made their performance better. If the words were given in the basic form (PlainBase), the results deteriorated 5.4 – 6.1 % units from results of PlainInfl. As the texts of the TUTK collection represent normal newspaper text, it can be argued that they also represent pretty well factual text types available and searched in the web. Thus it seems that the present state of the web search engines, where no truncation or linguistic aids for variation of key words are available, will deteriorate search results for highly inflected languages quite severely.

4. Discussion and conclusion

We tested in this paper enhanced inflectional stems in IR with short, web like queries. Inflectional stems of Finnish, a morphologically complex language, were enhanced with regular expressions that contained either one character from each possible case ending after the stem or full case endings. Our results show that the proposed method does not improve the IR performance of inflectional stems, but it does not considerably worsen it either. It should also be noticed, that the original inflectional stem method, Stemma, outperformed normalization slightly with short queries on two relevance levels.

References

Airio, Eija 2005 (to appear). Word normalization and decompounding in mono- and bilingual IR. In: *Information Retrieval.*

Alkula, Riitta 2001. From plain character strings to meaningful words: producing better full text databases for inflectional and compounding languages with morphological analysis software. In: *Information Retrieval* 4. 195−208.

Dougherty, Dale (1992). Sed & awk. Sebastopol, CA: O'Reilly & Associates, Inc.

Frakes, William 1992. Stemming algorithms. In: Frakes, W. B.; Baeza-Yates, R. (eds.) *Information Retrieval. Data Structures and Algorithms*. Prentice Hall. 131−160.

Friedl, Jeffrey 1997. Mastering regular expressions. Cambridge: O'Reilly & Associates, Inc.

Hollink, Vera; Kamps, J.; Monz, C.; de Rijke, M. 2004. Monolingual document retrieval for European languages. In: *Information Retrieval* 7. 33–52.

Jansen, Bernard J.; Spink, A.; Sarasevic, T. 2000. Real life, real users, and real needs: a study and analysis of user queries on the web. In: *Information Processing and Management 3*6. 207–227.

Jurafsky, Daniel; Martin, J.H. 2000. Speech and language processing. Low Price Edition. India: Pearson Education.

Kekäläinen, Jaana 1999. The effects of query complexity, expansion and structure on retrieval performance in probabilistic retrieval. Acta Universitatis Tamperensis 678.

Kettunen, Kimmo; Kunttu, T.; Järvelin, K. 2005 (to appear). To stem or lemmatize a highly inflectional language in a probabilistic IR environment? In: *Journal of Documentation* 61.

Kraaij, Wessel 2004. Variations on language modeling for information retrieval. CTIT Ph. D. series No. 04-62. Haag.

Partee, Barbara H.; ter Meulen, A.; Wall, R. E. 1993. Mathematical Methods in Linguistics. Corrected second printing of the first edition. Dordrecht: Kluwer Academic Press.

Porter, Martin F. 1980. An algorithm for suffix stripping. In: *Program* 14. 130−137.

Porter, Martin F. 2001. Snowball: A language for stemming algorithms. Retrieved November 28, 2003, from http://snowball.tartarus.org/texts/introduction.html.

Roche, Emmanuel.; Schabes, Y. 1997 (eds.). Finite-State Language Processing. Cambridge: The MIT Press.

Sormunen, Eero 2000. A method for measuring wide range performance of Boolean queries in full-text databases. Acta Universitatis Tamperensis 748.

Sproat, Richard 1992. Computation and morphology. Cambridge: The MIT Press.

KIMMO KETTUNEN is a senior assistant of language technology at the University of Tampere's Department of Information Studies. His research interests include linguistic tools for information retrieval of Finnish documents, linguistic aspects of IR and language technology in general. He has master's degree in general linguistics from the University of Helsinki and is a Ph.D. student at the University of Tampere. E-mail: lokike@uta.fi.

TERM EXTRACTION FROM LEGAL TEXTS IN LATVIAN

Valerijs Krugļevskis*, **Ilze Vancāne****
*Translation and Terminology Centre, Riga (Latvia),
**European Central Bank, Frankfurt am Main (Germany)

Abstract

Methods for term extraction from legal texts in Latvian have been compared. It has been shown that a linguistic method based on morphosyntactic analysis is more appropriate for computerised retrieval of terms in Latvian texts than the statistical method. Some new disambiguation procedures are presented. The importance of identifying direct objects in the sentences has been demonstrated. The paper also includes an analysis of the noun phrases obtained. Possible applications of the lists of extracted terms and the applicability of the method to other synthetic languages is also discussed.

Keywords: morphosyntactic analysis, disambiguation, terminology databases

1. Introduction

In Latvia like in other countries of Central and Eastern Europe in the period before accession to the European Union the translation of legal texts became a task of national significance. For many political, legal and economic and technological concepts new terms had to be coined, and there is an ongoing need for efficient terminology management tools well-adapted for the Latvian language.

At present, the most widely used translation method is computer-aided human translation. Usually, terms are stored in terminological databases, while translation memories are helpful both for retrieval of concordances or of those terms which for any reasons had not been entered into the database. The advantage of a database is that contexts and definitions and other comments, if necessary, are provided in its records.

However, it should be noted that both translation memories and terminological databases have the disadvantage that the information stored there is not always up-to-date. In some cases when several equivalents in the target language correspond to a term in the source language, it is not easy to find the synonym most appropriate for a specific translation.

The advantage of translation memories is that, apart of their main objective to ensure uniform translation of repeating phrases, almost every word and expression which has ever appeared in a previous translation can be found there. The drawback is that, unlike a database, no explicit comments can be found there, and that one sees the units of the final version of the text in target language only in the case when the final revision has been done using Translation Workbench in the same institution. If, for, example, the translation and final revision were performed in different places, it is highly probable that final corrections would not be included in the translation memory.

Since the primary objective of a translation process is to produce a precise translated text, and the main effort of translators and revisers is to ensure the quality of the final version, it would be quite natural to consider the final version of the text in the target language together with the original text as an important source of terminology and terminological expressions. Actually, a mechanism should be created to speed up the transfer of terms from the final version of the translated text into the database. The present work is aimed at facilitating this process.

The majority of articles dealing with term extraction methods suggest such procedures for texts in analytical languages. A term extraction procedure for an inflected language based on a linguistic method was developed for texts in Slovene (Vintar 2004).

To our knowledge our work is the first attempt to develop a procedure for extraction of terms from texts in Latvian.

2. Outline of the method

While there are satisfactory methods for extraction of terms from texts in analytical languages, these standard tools are inadequate for synthetic languages. It is important that even the truncation of endings which is used for search of declinable words in translation memories not always (short words, multiword terms) cannot produce satisfactory results.

Statistical term extraction methods are not applicable without modification to Latvian texts, because many software tools are not based on statistics of lemmas, but actually, on statistics of word forms. Thus, the number of occurences of a lemma is replaced by the number of occurences of its forms, and, as a consequence, the relative occurences of indeclinable parts of speech are largely exaggerated.

In this paper a linguistic method for extraction of terms from Latvian texts is described. This method is based on morphosyntactic analysis of the sentence aimed at localising multiword terms in the text. Usually, multiword terms are noun phrases, i.e., nouns with modifiers expressed as different parts of speech. In Latvian, these parts of speech may be adjectives, participles and nouns in the genitive. In Latvian multiword terms can also be expressed as more complicated syntactic structures. For instance, the modifier can be expressed as a noun preceded by a preposition. A detailed analysis of the syntactic structure of multiword terms has been carried out by V. Skujina (Skujina 2004). The authors have used the Multiterm terminology database with approximately 90,000 entries to investigate other possible syntactic structures of multiword terms in Latvian.

The term extraction procedure may be divided into the following steps.

The first step is the formatting of the text. A monolingual text is simply split in units which correspond to sentences in the text and to cells in a table. If it is expected to find the equivalents of the extracted terms in the target language, the texts in source and target texts have to be aligned, unless a fragment of the translation memory is used..

The second step is the morphological analysis of words in a sentence (a morphological database is used in order to determine the parts of speech and their morphological features).

The third step includes disambiguation procedures for homoforms.

The fourth step is the treatment and filtering of the output term list.

Since no semantic analysis is performed, the output list contains term candidates, and some words belonging to the general purpose language may also appear on the list.

3. Morphosyntactic analysis and disambiguation

In our previous paper dealing with computerised morphological analysis main principles were described and several disambiguation techniques were presented for legal texts in Latvian. Our methodology of disambiguation is based on the techniques described in one of our previous works (Krugļevskis, Vancāne 1994), but some improvements recently introduced will be mentioned.

The first algorithm for computerised morphological analysis was suggested in the 1990s (Greitane 1994). In our work a similar algorithm is applied. The stem of the word is looked up in the morphological database where stems are characterised by a code number that denotes the part of speech and the corresponding paradigm. Besides, the ending of the canonic form and the other stems corresponding to the same lemma can be found in the record. In Latvian, as a rule, a paradigm can be characterised by a specific stem and a set of endings.

The given word form may correspond to a) only one lemma, b) to several lemmas, c) to none (the relevant word has not been included in the morphological database). In the last case it is easy to add a new noun or adjective to the morphological database. The program produces an output file with the missing words. It suffices to put nouns belonging to the first to fifth declension into the nominative singular to enter them into the morphological database. The nouns of the sixth declension and adjectives have to be supplied with the appropriate paradigm numbers.

If the morphological features of certain words are ambiguous, the syntactic functions of the possible parts of speech have to be checked. In Latvian a noun may be preceded by an adjective or by a declinable pronoun. There is agreement in case, gender and number between the modifier and the headword. But there are also syntactic properties that characterise nouns in specific cases or numbers. Such properties can be used to determine the case of a homoform common to several cases. For example, the genitive singular of nouns belonging to the fourth and fifth declension coincides with the nominative plural and the accusative plural, and a further syntactic analysis is needed to determine the case of this word form.

The following rules are helpful to establish the case of some homoforms.

The genitive singular is either preceded by a preposition or followed by a noun in an arbitrary case, in contrast to a noun in the nominative plural.

The pre-modifiers of a noun in the accusative plural cannot be used to determine the case, because their endings are similar to the endings of the noun, and, consequently, they have also to be disambiguated. However, if in the morphological database transitive verbs have special indications, the accusative can be easily recognised, since a transitive verb in legal texts, usually, has a direct object, although such a grammatical requirement does not exist in Latvian.

A noun in the nominative can be either a subject of the sentence or of a clause, or can be an element of the compound nominal predicate. If such syntactic function is in conflict with the punctuation marks, that is, two subjects in the same sentence are not separated by a conjunction or comma, an alternative case must be selected.

A similar approach can be applied to finite verb /noun homoforms, because a finite verb can have only the predicate function in the sentence.

The conditions regarding the agreement with the modifiers are sufficient, but not necessary, since in Latvian there are no articles, and modifiers expressed as adjectives, as declinable participles or pronouns are optional. The other difficulty is the ambiguity of a word which appears adjacent to another word which has to be disambiguated.

However, the percentage of ambiguous word forms in Latvian is much lower than in English (in average, each eighth or ninth word in Latvian), and the probability of two consecutive ambiguous word forms could be estimated as approximately 1,5%. But even this situation can be resolved in many cases applying consecutively several disambiguation rules to each one of the words.

4. Discussion of the results

The output of the program is a list of term candidates. Since the translation is done by human translators, the process as a whole is not fully automatic, and the aim of the present method is to minimise the human labour in the term extraction.

A typical structure of a noun phrase could be similar to the following one:

(1) No(g)…Ad(g)..No(g)No(g)…Ad(n)No(n)

where No stands for a noun, Ad for an adjective and g and n in the parentheses denote the genitive and the nominative case. A sequence of nouns in the genitive or a sequence of adjectives and nouns in many cases can be decomposed into shorter noun phrases. For example, if the following noun phrases appear in the list of term candidates:

(2) *Eiropas Centrālo banku sistēmas Statūti* (Statute of the European System of Central Banks)

(3) *Eiropas Centrālās bankas Statūti* (Statute of the European Central Bank)

the computer program can establish that *Eiropas Centrālo Banku sistēma* and Eiropas Centrālā Banka are independent terms (appellations), but *sistēmas statūti* is not a term, because only the first two multiword terms appear as maximum noun phrases in the other articles of the statute.

A sequence of nouns in the genitive or a sequence of adjectives and nouns in many instances can be decomposed into shorter noun phrases. Some of this "sub-phrases" can be terms while other ones not corresponding to a specific concept may be considered to be phrases of the general purpose language or, in some cases, merely incidental sequences of words (for example, *sistēmas statūti).* The selection of terms from the list of term candidates can also be done manually by a terminologist.

At this stage the computer still can do a part of the work. Usually laws and regulations are structured in such a way that each section or article deals with a specific issue, and only terms describing very general and well-known concepts would occur in several sections of the same document. So the probability of identifying "sub-phrases" which are multiword terms by finding them in isolation in the same text sometimes may be very low. But the computer program can search the "sub-phrase" in isolation in large corpora containing legal documents. Thus, in many cases a program can perform an automatic decomposition of long noun phrases in to terms.

Another category of components are phrases and nouns belonging to the general purpose language. In legal texts the percentage of such words is not high, and a file of general purpose words which may appear in legal texts can be compiled to exclude some of them from consideration.

If term extraction is a part of a translation process, it is important to emphasise that, usually, a translator with some experience is more interested to find terms that do not occur frequently in the text, and that is also an advantage of the linguistic method that terms with few occurrences are not automatically discarded.

In some circumstances it is important to achieve consistency of terminology between two specific documents. In such case the program can produce a list of term candidates common for both documents. If aligned texts are available, the application of the suggested method will show immediately which terms and their equivalents should be checked.

Since manual work in the term extraction can be only reduced, but not eliminated, it is important to make automatic as far as possible all other operations. For example, a program has been developed to enter terms form Word tables into a Multiterm database, and the terms already stored in the database are automatically excluded from the list of term candidates. It is thereby possible to reduce the number of term candidates by as much as 50%.

The structure of noun phrases is language specific, but many features of the procedure are common to several East European languages. The procedure of finding the canonic form of a word could be very similar in Slavonic and Baltic languages as well as disambiguation based on agreement of modifiers and headwords.

Although the development of a procedure for term extraction from Latvian texts is only in its initial stage, this procedure has already been helpful to achieve consistency of terminology in long documents translated by external translators. The next improvement in this procedure would be development of at least a partially automatic retrieval of term equivalents in the target language from aligned texts.

References

Greitāne, Inguna 1994. Latviešu valodas lokāmo vārdšķiru locīšanas algoritmi. In: *Latvijas Zinātņu akadēmijas Vēstis*, No. 1, 32–39.

Krugļevskis, Valerijs; Vancāne, Ilze 2004. Computerised morphological analysis and related disambiguation problems in legal texts in Latvian. In: *The first Baltic conference. Human language technologies – the Baltic perspective*, April 21-22, Riga, Latvia, 81–84.

Skujiņa, Valentīna 1993. Latviešu terminoloģijas izstrādes principi. Rīga: Zinātne.

Vintar, Špela 2004. Extracting terms and terminological collocations from the ELAN Slovene-English parallel corpus. Retrieved February 28, 2005 from http://nl.ijs.si/eamt00/proc/Vintar.pdf

KRUGĻEVSKIS, VALERIJS is a terminologist at the Translation and Terminology Centre (Riga, Latvia) since 1998. He has a doctoral degree in physics. V.Kruglevskis has worked at the Institute of Mathematics and Computer Science, University of Latvia, as a research scientist from 1984 to 2000. E-mail: valerijs.kruglevskis@ttc.lv

VANCĀNE, ILZE (1974), Latvia. Mg. Phil. (Latvian language), University of Latvia (LU). 2001–2004 Ph.D. studies at the LU. Currently she is a terminologist/ language technologist at the European Central Bank. Her research interests include translation-oriented terminology research methods and principles, terminology and translation work automation. Author and co-author of 11 articles and conference proceedings.

DERIVING PITCH ACCENT CLASSES USING AUTOMATIC F0 STYLISATION AND UNSUPERVISED CLUSTERING TECHNIQUES

Dominika Oliver
Institute of Phonetics, Saarland University (Saarbrücken, Germany)

Abstract

This paper presents a method for describing pitch accent classes in Polish by means of automatic F0 stylisation and unsupervised clustering methods. Precise parameters of accent classes are generated from an automatically stylised F0 curve, obtained using a Momel algorithm augmented for boundary locations. The revised algorithm for these locations has been tested in a perceptual study which shows that subjects judge the sentences stylised with the revised Momel to be perceptually equivalent to the original signal and at the same time better than the original algorithm. The parameters obtained from the stylised curve are used as input to Kohonen self-organising maps and a hierarchical clustering technique. The results identify three classes of pitch contours present. Utterances automatically labelled using this classification can be used as input to a prosody prediction and generation module of a speech synthesis system.

Keywords: acoustics, cluster analysis, data mining, hierarchical agglomerative clustering, intonation modelling, Momel, Polish, prosody, Prosogram, speech resources, Self-Organising Map

1. Introduction

In order to obtain a high quality speech synthesis, a good prosodic modelling needs to be developed. One of the steps in modelling prosody, apart from predicting duration and phrasing, is to predict and generate appropriate pitch accent contours. To train an accent type prediction system, it is necessary to have a good typology of accent types present in the language. Moreover, we need precise parameters corresponding to each category, which can be derived directly from the F0 curve if a prosodically pre-labelled database is available. However, the non-trivial, prosodic labelling introduces human annotator bias, especially when the annotation scheme used is based on phonological categories (e.g. ToBI). The resulting annotation is then partly phonological and phonetic, which makes the task difficult task for a machine learner.

In this study, we aim to derive prototypical pitch contour types found in Polish, based on their acoustic characteristics. For classification we use Kohonen Self-Organising Maps (Kohonen 1995), a method which has been applied in prosodic event classification e.g. (Werner 2001), and which does not presuppose a number of clusters. The resulting classification enables an automatic re-annotation of the database with intonation events grouped according to similarity of acoustic parameters. This way, any method responsible for prediction and generation of different pitch accents will be equipped with a consistent specification of pitch accents.

This paper is organised as follows: section 2 describes the speech resources used for the study. Section 3 deals with F0 stylisation methods and describes our modifications made to the Momel method followed by a perceptual study. In section 4, we detail pitch accent clustering analysis, where we introduce the parameterisation and clustering techniques applied, followed by results and conclusions in section 5.

2. Resources

The current study uses the PoInt speech database of Polish (Karpiński and Kleśta 2001), which has been recorded as part of the Polish Intonation Database Project. This database contains a variety of discourse types: fragments of read literary texts, quasi-spontaneous monologues, as well as map task based dialogues. In order to avoid accent truncation and speech overlaps present in the map task material, we use a subset of the database, leaving out spontaneous speech utterances. The database contains of recordings of male and female speakers and was phonetically and prosodically annotated on the syllabic level. Additionally, a Polish part of the Babel database (Gubrynowicz 1998), consisting of recordings of read passages, was used.

3. F0 stylisation

There are a number of methods used to stylise and model a fundamental frequency curve; purely acoustic and those based on human perception of pitch. An example of an acoustic method, which will be used in this study, is Momel (Hirst and Espesser 1993). This F0 stylisation method is based on the technique called asymmetrical modal quadratic regression. The algorithm models the macroprosodic component of F0 as a quadratic spline function resulting in a continuous contour. A perceptual approach is used in the method proposed by d'Alessandro and Mertens (1995), where stylisation is based on a model of tonal perception by humans and takes the syllabic nucleus as a basic unit. The F0 contour is transformed into a sequence of static and dynamic tonal segments. This perceptual method takes into account three factors, namely, segmentation into syllabic or vocalic nuclei, the glissando threshold and the differential glissando threshold.

3.1. Momel modification

The quantitative and qualitative evaluation of The Momel F0 stylisation algorithm for French and Italian as reported by Campione and Véronis (2000) reveals three types of systematic errors: target points in wrong positions, redundant target points and missing target points. The third group of error (missing points) can be divided into two types; those occurring before a pause and those occurring after a pause. As a result, a final contour can be misrepresented in such a way that a final fall is replaced by a final rise, and, in the second case, utterance initial stylised F0 values are too high for a particular speaker.

The modifications we propose to the Momel algorithm pertain to the third group of error, namely, we augment its behaviour in boundary conditions. In order to do this we need to identify reasons why the algorithm is prone to error before and after the pause and what measures have been taken by the authors of the algorithm to correct it. When there is a concave curve at the end of an utterance, the algorithm will only detect one target point where intonation changes its direction. Then, a target point at the same vertical and horizontal distance from the end of voicing as the end of voicing is from the previous target is calculated (Hirst, personal communication). The assumption behind it is that if a target point is put there, the stylised curve will capture the final rise.

The solution is not successful in all cases and we propose an alternative treatment of the problem. We do not calculate additional target points as suggested above. Instead, we take away the first and last Momel point computed according to the above strategy, as well as Momel points not in the speaker's range. In order to span the whole signal we add, if present, original F0 points to the Momel target points at the boundary locations. In cases of unvoiced material at both ends of the signal, the values before the F0 onset and after the F0 offset are replaced, as used by Mixdorff (2005) in the preprocessing stage for Fujisaki-model parameters extraction, with the first and last F0 value, respectively, found in the extracted contour. In this way elements of the stylised curve are enriched with the initial and final variations of F0 essential for capturing boundary prosodic events. The new continuous F0 curve is then derived by quadratic interpolation of the Momel target points and original F0 points added at the beginning and end of an utterance.

3.2. Perceptual study

In order to test if F0 stylisation using the modified Momel method is perceptually equivalent to the original sound and can thus be used for deriving parameters of prosodic events, a human perceptual study in form of a rating test was carried out. To judge the modified algorithm's suitability, the task was to compare sentences stylised with the original Momel, the modified Momel and a perceptually based method, Prosogram (d'Alessandro and Mertens 1995), with the original sentence. The hypothesis was that the modified Momel algorithm would be perceptually equivalent to the original sentence and be rated higher than the original Momel. At the same time, the ratings of our augmented algorithm should not differ significantly from Prosogram. The method used in the study is an enhanced Visual Analogue Scale called the Visual Sort and Rate method (Granqvist 2003) and Rating Test software (Schröder 2004). A total of 15 participants, both male and female native speakers of Polish rated three resynthesised versions of audio stimuli, ranging from 0.96 to 3.78 sec., representing three F0 stylisation methods. The subjects were always asked to rate the similarity of three target sentences to one main stimulus. The main stimulus was an original sentence from the database which has also been resynthesised. The three target stimuli were versions of the main stimulus and differed from it only in the way their F0 contour was stylised.

3.3. Results

Figure 1 shows inter-subject rating scores for sentences synthesised using Prosogram, the original Momel method and the modified Momel stylisation method. Modified Momel target sentences were rated as significantly closer to the main sentence than those stylised using the original Momel algorithm. The rating of Prosogram is overall the highest of all methods but does not significantly differ from the modified Momel ratings (receiving mean values 70 and 67 out of 100, respectively). Both the results of the practice part and test proper are consistent and are not significantly different, thus show no learning effect.

There is a significant difference in the rating of tokens coming from the two databases, i.e. Babel sentences, which were segmented into phonemes, receive better ratings than PoInt which is only syllabically segmented ($p<0.001$). The difference in rating is visible in the scores for Prosogram stylised stimuli. This is easily explained by the nature of Prosogram, which relies on syllabic nuclei identification. Polish syllables in stimuli contain complex consonantal material in the syllable onset, which is often voiceless, thus impeding the algorithm. The same material, namely four stimuli which were rated systematically worse than both the Momel versions when presented as input with a phonemic annotation to the same program, yield significantly better resemblance to the original sentence than those based on syllabic segmentation (p<0.001). Prosogram finds, on average, six times more stylised target points in this condition.

We conclude that, when faced with speech material without segmental annotation, the stylisation using modified Momel can be considered a fine method for representing F0 curve, yielding results approaching the ratings of a perceptual method Prosogram. Moreover, the sentences stylised using both of these techniques are viewed as being perceptually close copies of the original speech signal.

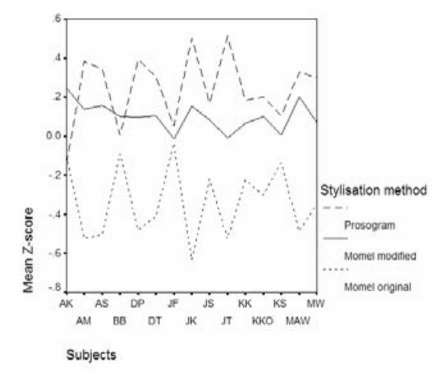


Figure 1: Inter-subject rating scores

4. Pitch accent clustering analysis

Based on the results of the perceptual study, the F0 contours stylised using the modified Momel method are used for the extraction of parameters in accent categorisation. The goal of the study is to find prototypical phonetically distinct pitch accent classes, without the need for a priori knowledge about their phonological categories. For our exploratory approach, a two stage procedure is used. First, we apply Self-Organising Maps (SOMs) (Kohonen 1995), a vector quantisation method which performs clustering of instances, in the form of acoustic feature vectors. It determines the most similar prototypical instance, and then adjusts its coordinates so that it becomes even more similar to the training example. As a result, we obtain similarity clusters that can be seen as soft edged classes or fuzzy sets emerging from statistical correlations. To derive the final number of clusters from the resulting topological representation of the original data, hierarchical agglomerative clustering is applied. By cutting the resulting dendrogram in the place where there is a large distance between two clusters and merging the clusters based on their proximity, we determine the final number of clusters. The main motivation behind the proposed algorithms is not specifying but discovering how many pitch accent classes would describe the intonational variation present in the database used for training (i.e. read and semi-spontaneous speech).

Since the corpus used consisted of both male and female speakers, we normalised the values by first converting them to semitone values and z-scores calculated per sentence, and scaling to the mean of the database. Based on the prosodic annotation marking the positions of pitch accents within the utterance, the stylised contours have been parameterised. For the description we used the alignment of F0 peak with respect to syllable start, the form of movement comprising movement slope and the pitch amplitude.

An ANOVA analysis determined that these parameters do not vary significantly between male and female speakers in the database.

4.1. Results

The clustering performed using SOMs and hierarchical clustering, based on the above parameters, results in three clusters. Figure 2 shows the prototypical contours found in each cluster associated with the accented syllable.

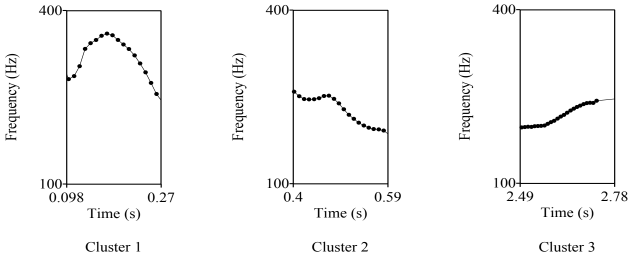


Figure 2: Prototypical contours in clusters 1-3

Cluster 1 is the smallest cluster, containing 200 tokens (16.1% of the total pitch accents). The contours in this group exhibit the steepest rising and falling movement, with F0 peak positioned in the middle of the syllable. These contours reach the highest F0 maximum values and the lowest F0 minimum values of all clusters. 20% of members in this group are phrase non-final pitch accents and mostly associated with emphatic focus or a wh-word in a question. Cluster 2 contains 398 tokens (32% of the total pitch accents). This group is characterised by a very early F0 peak, mostly starting at the syllable start or early in the onset forming a plateau, followed by a gradual fall ending at the speaker's mid or bottom range. 90% of the pitch accents in this cluster are phrase final and are characteristic of a final fall. Cluster 3 is the largest cluster, containing 51.1% of the pitch accents (642 tokens). The pitch accents in this group exhibit a very late F0 peak, either in the syllable offset or on the following syllable. This type of contour is found mostly in phrase final position and is typically associated with a continuation rise.

5. Conclusions

As the speech material used in this study does not cover all aspects of speech variations (e.g. expressive speech), it does not represent a complete intonation model of Polish. Nevertheless, it is considered sufficient to propose collapsing the number of accent types to be used for intonation generation to three types. The analysis shows that automatic classification can be successfully applied for the task of deriving pitch classes if a perceptually equivalent stylisation of F0 is carried out. The results of a perceptual study shows that by augmenting an existing stylisation method Momel we obtain an F0 curve, which is found to be perceptually equivalent to the original signal. From an automatically stylised F0 curve precise parameters of accent classes can be generated. Furthermore, using unsupervised clustering techniques which do not require a pre-determined number of categories is suggested.

Self-organising maps are, thus, considered a reasonable method for accent categorisation and their results linguistically meaningful and consistent. Features, like F0 peak delay and slope, seem to provide sufficient prosodic information for such classification. The derived pitch accent classes, parameterised in this way, can be used for pitch accent modelling. As these results are based on statistical analysis they need to be further tested in a human perceptual study.

6. Acknowledgements

This study was funded by the International Postgraduate College "Language Technology and Cognitive Systems". The author wishes to thank Daniel Hirst, Piet Mertens, Guillaume Rolland, Stefan Werner and Magdalena Wolska for discussions and technical help.

References

Campione, E.; Véronis, J. 2000. Une évaluation de l'algorithme de stylisation mélodique momel. In: *Travaux Interdisciplinaires du Laboratoire Parole et Langage d'Aix-en-Provence* **19**, 27–44

d'Alessandro, C.; Mertens, P. 1995. Automatic pitch contour stylization using a model of tonal perception. In: *Computer Speech and Language* **9(3)**, 257–288

Granqvist, S. 2003. The visual sort and rate method for perceptual evaluation in listening tests. In: *Logoped Phoniatr Vocol.* **28(3)**, 109–116

Gubrynowicz, R. 1998. The Polish Database of Spoken Language. In: *Proc. LREC*: Vol. 1, Granada. 1031–1037

Hirst, D. J.; Espesser, R. 1993. Automatic modelling of fundamental frequency using a quadratic spline function. In: *Travaux de l'Institut de Phonétique d'Aix* **15**, 75–85

Karpiński, M.; Kleśta, J. 2001. The project of an intonational database for the polish language. In: *In Prosody 2000*, Adam Mickiewicz University: Poznań

Kohonen, T. 1995. Self-Organizing Maps: Vol. 30 of *Springer Series in Information Sciences*. Berlin, Heidelberg: Springer-Verlag

Mixdorff, H. 2005. Fujisaki Parameter Extraction Environment. Retrieved January 20, 2005, from http://www.tfh-berlin.de/˜mixdorff

Schröder, M. 2004. Speech and Emotion Research: An overview of research frameworks and a dimensional approach to emotional speech synthesis. *Ph.D. thesis*: Saarland University

Werner, S. 2001. Acoustic classification of intonational events. In: *In Prosody-2001*

DOMINIKA OLIVER is currently a PhD student within the International Postgraduate College "Language Technology and Cognitive Systems" in the Institute of Phonetics at Saarland University. She received her MSc. (Speech and Language Processing) at the University of Edinburgh, having developed a Polish text-to-speech synthesis module in Festival TTS system. Her research interests include speech synthesis and intonation analysis with an emphasis on Polish. Her PhD work deals with the modelling of Polish intonation for speech synthesis.

AN EFFICIENT ONE-PASS DECODER FOR FINNISH LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

Janne Pylkkönen
Neural Networks Research Centre, Helsinki University of Technology, Finland

Abstract

This paper describes a design of a one-pass large vocabulary continuous speech recognition decoder aimed for Finnish. The decoder is based on the popular time-synchronous beam search approach, extended to handle some language dependent issues. For the construction of the static part of the search network a new algorithm is presented, which enables efficient use of shared state HMMs. The search network also includes statically expanded cross-word triphone contexts, which increase the memory requirements only modestly. The beam search is enhanced with a bigram language model look-ahead technique, implemented using simple table lookups and an efficient caching scheme. Compared to our previous decoder, the new design achieves 24% relative reduction to phoneme error rate, with a near real-time performance.

Keywords: static search network, cross-word triphone contexts, language model lookahead

1. Introduction

The continuing increase in the complexity of acoustic and language models used in speech recognition has imposed growing requirements to the efficiency of decoders. Especially the introduction of cross-word acoustic models and long-span language models has led to development of many new solutions for the decoding problem. Compared to many established solutions in the field of speech recognition, the selection of different decoding techniques is relatively broad. A good overview of these is given in (Aubert 2002).

To promote the simplicity of design, we have chosen to use the one-pass time-synchronous approach. The problem is then how to build an efficient search network, and which knowledge sources it should contain. Probably the most fine-tuned solutions for the search network optimization are the WFST methods (Mohri et al. 2002). The idea is to combine all the knowledge sources together statically, and optimize the network for maximal efficiency. This, however, can restrict the complexity of the knowledge sources, and prevent some on-the-fly adaptation. A completely opposite solution is to expand the search network dynamically (Sixtus and Ney 2002). This, on the other hand, may be computationally too expensive for efficient decoding.

The architecture adopted for our decoder is in between these two network expansion methods. Similar to the approach presented in (Demuynck et al. 2000), the lexicon and acoustic models are combined into a static network, and the language model is used dynamically during the search. In this paper, we present the search network developed for our decoder, and also a novel language model lookahead scheme used to enhance the beam search. The target language has also imposed requirements to the decoder. Some problems in Finnish large vocabulary speech recognition (LVCSR) are highlighted and solutions to them are presented.

2. Language dependent issues

Finnish is a highly inflectional compounding language, so the number of distinct word forms in everyday use is very large. It is therefore very difficult to construct a good n-gram model based on words as lexical units. Instead of words, we use morpheme-like sub-word units called morphs which are discovered from a large corpus in an unsupervised manner (Siivola et al. 2003). The size of the lexicon can then be relatively small, although the vocabulary of the recognition is virtually unlimited. In the system presented here, there are about 26000 morphs in the lexicon.

In decoder, the use of sub-word units implies that the word boundaries are not obtained automatically. The word boundaries are resolved by duplicating the paths with possible word boundaries, and inserting the word boundary to one of these paths. If there is acoustic silence between the morphs, the word boundary is always inserted.

From now on, to maintain the common terminology, the words “word” and “morph” are used interchangeably. Therefore, for example, the term “word history” should in this context be “morph history”, but the former is still used for clarity.

Another issue in Finnish speech recognition is that most of the Finnish phonemes have two length variants, distinguished from each other by different durations of the phones. To improve the discrimination between these phoneme variants, the HMM state durations are modeled explicitly using gamma distributions. In decoding, the duration probabilities were added to the total likelihood, similar to the post-processing approach in stack decoders (Pylkkönen and Kurimo 2004). As the acoustical differences between the phoneme variants are small, these phonemes are combined in triphone contexts so that the two variants are distinguished only in the central phoneme of the triphone.

3. Constructing the search network

In (Demuynck et al. 2000), a procedure for constructing a compact search network was presented. Here, a new and simpler scheme is presented, which still achieves a relatively compact representation of the constraints set by the lexicon and acoustic models. Instead of first building a context independent network, most of the search structure is constructed at once, and only minor post processing is required.

The search network is built around a popular idea of a lexical prefix tree. As suggested by Demuynck et al. (2000), the traditional phone-level tree can be made even more efficient by utilizing the HMM level state tying, which has been implemented here. Cross-word triphone contexts are handled by building a separate network, to which the lexical prefix tree is linked. The new network structure is very compact, and increases the size of the search tree only moderately.

We use the following terminology. The search network is built of *nodes*, which are linked to each other with *arcs*. Nodes can either correspond to one *HMM state*, or be *dummy nodes* without any acoustic probabilities associated with them. In decoding, these dummy nodes are passed immediately, they merely mediate the *tokens* used to represent

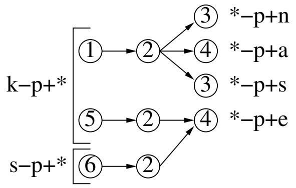


Figure 1: An example of the fan-in triphone node organization. The numbers inside the nodes represent the (shared) HMM state numbers. Triphone notation k-p+* represents /p/ after /k/ and followed by any phoneme.

the active search network. A node can also have a *word identity* associated with it, which leads to insertion of the word into the word history of the token passing that node.

Currently the construction procedure assumes triphone models, but it could be expanded to handle also wider contexts. Every triphone is defined in the acoustic models, and they are not tied at the triphone level. Instead, each triphone has a set of HMM states (currently three states in a left-to-right topology), and these states are shared among all triphones. The state tying has been performed using a decision tree.

Next, the steps in the construction of the search network are discussed in detail.

3.1. Creating fan-in triphones

The construction of the search network starts by creating the fan-in triphones. This means that the HMM state sequences of every triphone are inserted to the search network. They are organized by their central phoneme and the left context, so that if these are equal, the fan-in triphones are allowed to share their first nodes. The last nodes, however, are grouped differently, now according to the central phoneme and the right context. This way the number of arcs are minimized when linking other nodes to or from the fan-in triphones. An example of the resulting node connections is shown in Figure 1.

3.2. Construction of the lexical prefix tree

The lexical prefix tree is constructed by adding words to the tree one at a time. The words are first expanded to the corresponding HMM state sequence. The construction algorithm then starts from the (dummy) root node, and finds the path in the tree which is common to the given state sequence. When a node is reached from which the path can not be continued, the rest of the states are inserted as tree nodes starting from the last common node. For the first phonemes, the left context is silence, so in decoding the root node is accessed only after silence. Other contexts are handled by fan-in triphones, which are linked to the second phonemes of the words, as explained in the next subsection.

The states are added up to the second last phoneme of the word. From there an arc is made to a dummy node containing the word identity of the new word. This node is then linked to proper fan-out triphones, which are created on demand.

The fan-out triphones are organized the same way as the fan-in triphones so that triphones belonging to the same phoneme and having the same left context are grouped together and are allowed to share their common states. When the dummy node at the end of a word is to be linked to the fan-out triphones, the node is linked to every starting node of the corresponding group (defined uniquely by the last triphone of the word).

Single phoneme words must be handled separately. The only node linked to the root node is the dummy node with the word identity. This is linked to the fan-out triphones

Table 1: Search network statistics

	nodes	arcs
Fan-out	13162	60572
Fan-in	22471	84693
Prefix tree	198829	451295
Total	234462	596560

as explained above. However, for single phoneme words also another implementation of the word has to be added inside the cross-word network. This is done by adding a dummy node with the word identity after every fan-in triphone whose central phoneme corresponds to the given word. This dummy node is then linked back to the fan-in triphones, determined by the right context of the originating fan-in triphone.

3.3. Building the cross-word network

After the words have been added to the prefix tree and the fan-out triphones have been created, the last nodes of the fan-out triphones are linked to the corresponding fan-in triphones. This simply means adding arcs to the first nodes of the corresponding group of fan-in triphones. Remember that each fan-out triphone has their last nodes shared with the triphones having the same central phoneme and the right context. The corresponding group of fan-in triphones is therefore determined uniquely, and the number of arcs is minimized.

The fan-in triphones are then linked back to the lexical prefix tree to make the tree re-entrant. This is done similarly as linking the fan-out triphones to fan-in ones, but now the targets are the nodes belonging to the first states of the second phonemes of the words.

3.4. Post processing

In a post processing phase, the word identities are propagated towards the root of the node, to obtain unique word identity in decoding as early as possible. Also in post processing, the word identity tables for language model lookahead are collected.

Table 1 shows some statistics of a tree constructed using the 26k morph lexicon. There were 2165 different HMM states in 30000 triphones (48 central phonemes, 25 different left and right contexts). As can be seen in the table, the number of nodes in the cross-word network is quite small compared to the number of nodes in the prefix tree. These figures seem rather similar to the ones reported in (Demuynck et al. 2000), although it must be noted that the number of triphones and their tying can affect the structure of the search network greatly.

4. Language model lookahead

When the lexical network is constructed as a prefix tree, the word identities can be determined only after there are no more branches in the tree. Thus the inclusion of the language model (LM) probability is delayed. It is well known that using LM probabilities as early as possible enhances the beam pruning and therefore decreases the size of the search space. This can be achieved by applying a language model lookahead technique.

Ortmanns and Ney (2000) presented a method for computing LM lookahead scores using a compressed tree structure to propagate the scores of individual words to the nodes of the search network. In principle that method could compute the scores only for nodes for which the scores are needed. However, for our purposes, this kind of method seems to be overly complex, considering that to determine the language model scores at the beginning of the tree, all the nodes would have to be processed anyway. Therefore for our decoder, a simpler approach was adopted. After the lexical network has been built, a list of possible word identities which are reachable from each node are collected and saved to the nodes. The LM lookahead score can then be computed by finding the maximum of the LM scores over the words in the node's list.

To minimize the significant amount of redundant computations involved in the LM lookahead, we created a two level caching system. Each node contains a simple cache of the maximum LM scores of the possible follow-up words for different word histories. If the correct word history is not found in the cache, a higher level cache is referenced. It stores the LM scores of all the words for a certain number of previous word histories. In case of a cache miss, the probabilities of all the words in the LM for the given word history are computed and stored to the cache. With back-off language models, this computation can be done rather efficiently. During preliminary evaluation it was noted that for the LM lookahead bigram language model gave the best performance with respect to computational effort. For the final LM, a 4-gram model was used.

In (Ortmanns and Ney 2000) it was suggested that LM lookahead is used only in the first nodes of the lexical prefix tree. In a preliminary evaluation four node generations were found to be enough. The LM lookahead is also applied only in those nodes where the list of possible word identities has changed from that of the previous nodes. Reducing the number of nodes in which LM lookahead is applied helps to save memory when a node level caching is involved. With four generations, the final number of nodes with LM lookahead caches was 2503, when without this reduction 17087 node level caches would have been needed.

5. Experimental evaluation

The new decoder design was compared against our previous decoder (Hirsimäki and Kurimo 2004), which was based on the principle of stack decoding. Also the effect of the cross-word (CW) network was analyzed by running the new decoder with and without the cross-word triphone handling.

The evaluation task was a Finnish speaker dependent LVCSR task (Siivola et al. 2003). Preliminary testing and parameter optimization was done with a separate 19-minute development set of the same material. The length of the actual evaluation set was 27 minutes. The acoustic models were trained from a training set of about 12 hours.

The results are shown in Table 2. We use phoneme error rate as the main error measure, as it better indicates the recognition performance of a highly inflectional language than the traditionally used word error rate. The efficiency was measured by a real-time factor, which does not include the time used to compute the framewise state probabilities.

It can be seen that the use of cross-word triphones improves the recognition accuracy by about 26%. This is much more compared to the improvements reported in English (Sixtus and Ney 2002). The reason for this is probably the sub-word lexicon, which gives rise to a much higher amount of cross-word contexts. It is also remarkable that the use of cross-word triphones actually makes the decoder even faster. This contradicts the effect of using cross-word contexts with dynamically expanded search space (Sixtus and Ney 2002), showing the efficiency of our static expansion of the search network.

Table 2: Results of the decoder evaluation

Decoder	Phoneme error	Word error	Real-time factor
Stack decoder	2.58%	15.5%	4.3
New, no CW triphones	2.65%	15.5%	3.1
New, with CW triphones	1.96%	12.6%	1.3

6. Conclusions

This paper presented the design of a decoder which is able to handle cross-word triphone contexts efficiently using a statically expanded search network. The decoder also includes some language specific extensions necessary for Finnish LVCSR. The decoder uses a one-pass time-synchronous beam search, which is enhanced using a language model lookahead technique, also presented in the paper. The decoder was compared against our previous decoder, and the performance proves the efficiency of the new design.

7. Acknowledgments

This work was supported by the Academy of Finland in the project “New adaptive and learning methods in speech recognition”.

References

Aubert, Xavier L. 2002. An overview of decoding techniques for large vocabulary continuous speech recognition. In: *Computer Speech and Language* 16, 89–114

Demuynck, Kris; Duchateau, Jacques; Compernolle, Dirk Van; Wambacq, Patrick 2000. An efficient search space representation for large vocabulary continuous speech recognition. In: *Speech Communication* 30, 37–53

Hirsimäki, Teemu; Kurimo, Mikko 2004. Decoder issues in unlimited Finnish speech recognition. In: *Proceedings of Norsig*. 320–323

Mohri, Mehryar; Pereira, Fernando; Riley, Michael 2002. Weighted finite-state transducers in speech recognition. In: *Computer Speech and Language* 16, 69–88

Ortmanns, Stefan; Ney, Hermann 2000. Look-ahead techniques for fast beam search. In: *Computer Speech and Language* 14, 15–32

Pylkkönen, Janne; Kurimo, Mikko 2004. Using phone durations in Finnish large vocabulary continuous speech recognition. In: *Proceedings of Norsig*. 324–327

Siivola, Vesa; Hirsimäki, Teemu; Creutz, Mathias; Kurimo, Mikko 2003. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In: *Proceedings of Eurospeech*. 2293–2296

Sixtus, Achim; Ney, Hermann 2002. From within-word model search to across-word model search in large vocabulary continuous speech recognition. In: *Computer Speech and Language* 16, 245–271

JANNE PYLKKÖNEN (M.Sc.) is a post-graduate student working as a researcher at the Neural Networks Research Centre of Helsinki University of Technology. E-mail: janne.pylkkonen@hut.fi.

LITHUANIAN SPEECH RECOGNITION BY IMPROVED PHONEME DISCRIMINATION

Vytautas Rudžionis, Kęstutis Driaunys, Pranas Žvinys
Vilnius university Kaunas faculty of humanities, Kaunas, Lithuania

Abstract

The improved phoneme discrimination still remains important issue for speech recognition task. Some recently performed experiments with Lithuanian continuous speech recognition showed that majority of the errors occur at the ends of the words. Generally there are several possible ways to achieve better recognition accuracy in such situation: to improve language model and to improve phonemic discrimination.

Here we present our attempts to improve phoneme discrimination via hierarchical phoneme discrimination approach. All phonemes are divided into a series of higher level classes (such as vowels, semivowels, fricatives, etc.) using MFCC based features and linear discriminant analysis. Later extended analysis of acoustic properties of phonemic unit is performed to allow proper discrimination of this phoneme in the selected subclass.

Experimental results showed that such approach was particularly effective for vowel discrimination. This is important since most of the continuous broadcast news recognition errors were observed on the last vowel of the word. Moderate increase in consonant recognition accuracy has been observed too.

Keywords: discriminant analysis, speech labeling, overall performance, discriminational power, phonemic recognition, HMM

1. Introduction

Speech recognition accuracy is crucial factor for commercial success of speech technology. Despite the fact that each year number of successfully deployed automatic speech enabled systems grows rapidly is not so good for speech recognition implementing systems. Most of the growth in speech technology systems is in the speech synthesis and speech compression implementing systems. Growth in the systems based on speech recognition as the core of the system is slower.

Most of the successes with speech recognition systems during last decade were related with the success of the continuous density HMM model and various modeling tools based on it. The implementation of HMM based speech recognition created basis for speaker–independent continuous speech recognition system development for various languages. Those systems could achieve about 90 percent recognition accuracy in average. Unfortunately such recognition accuracy level isn't enough for a vast majority of speech recognition applications.

Looking to the tendencies of speech recognition research during last three years we could observe some sort of stagnation in speech recognition system development. Most of research presents results that shows only moderate increase in recognition performance or even this increase is observed only in some predefined conditions. This could be caused by the fact that majority of authors still uses modifications of basic HMM approach which was investigated at large in previous years. So current situation in the speech recognition could be characterized as saturation of HMM model. Only very few researchers tries to look for another approaches that could be useful and could lead to progress in speech recognition area.

States of HMM or elementary HMM‘s in a chain makes attempt to simulate speech signal as a linear sequence of phonemes with additional auxiliary acoustic events (silence, noise, space, etc.) (Koizumi et al. 1996). In such systems phonemes don‘t utilize exact structure of acoustic-phonetic properties of signal except of traditional HMM topology (three emitting states, directed from left to right). Researchers are looking for alternative or HMM supplementing approaches. One of such approaches could be based on better discrimination of phonetic characteristics of speech signal.

Modern speech recognition systems do not exploit enough discrimination capabilities in the acoustic- phonetic level of speech signal. (Lipmann 1996) showed that capabilities of human auditory system and automatic recognition systems differs significantly, especially when recognizing sentences without sense (it means that recognition system can’t use any form of grammar). Despite that this study was conducted several years ago little has been done to improve phonemic automatic speech recognition during this period. Recently Lithuanian large vocabulary continuous speech recognition experiments (Silingas et al. 2004) showed that majority of recognition errors occurred at the last phoneme of the word. In this case error rate could be reduced using some sort of Lithuanian grammar model but there will be situations when only better discrimination in the acoustic-phonetic level could lead to proper recognition.

Our group for many years is searching to ways how to improve overall speech recognition via better discrimination in the acoustic- phonemic level of speech. We believe that this is perspective direction of speech research. In this paper we’ll try to present structural phoneme classification algorithm. It is based on the assumption that phonemes describing features has enough information to find and to exploit structural properties (specific for some class of phonemes) of speech signal. These features in the basic HMM are exploited insufficiently.

2. Linear discriminant analysis in phoneme based speech recognition

Linear discriminant analysis is one of the statistical classification methods with strongest discriminational characteristics. At the same time this is robust and convenient method for realization. Naturally these properties raised attention of researchers in various fields. There were numerous attempts to apply LDA for speech recognition tasks as well.

Various studies carried out during last decade showed promising results when LDA was applied to speech recognition tasks. Despite this it didn’t get wider interest. This could be caused by the difficulties rising from additional problems that need to be solved when LDA is applied and successful development of HMM based systems and consequent hopes from this model.

We used linear discriminant analysis in our previous research also. One of the successful applications of one of the linear discriminant analysis forms – regularized discriminant analysis – could be found in (Rudzionis 1999). But in current study we significantly expanded speech database used in experiments both in the sense of number of speakers and number of phonemes. Here we used Fisher linear discriminant function

– simplest form of linear discriminant analysis that is known of its efficient performance and robustness in various experiments and for various tasks:

$$J(w)=(x^1-x^2)S^{-1}(x^1-x^2)'$$

where: w – a priory class probability, x^1, x^2 – vectors of both classes means, S – joint covariance matrix of both classes.

Typically during recognition tested phoneme should be compared with all phonemes template. Here we try to apply hierarchical scheme: at first to classify tested phoneme into phoneme group and then to make decision inside this group. Four phoneme groups we used in experiments: vowels, semivowels, plosive and fricative consonants (Girdenis 1995).

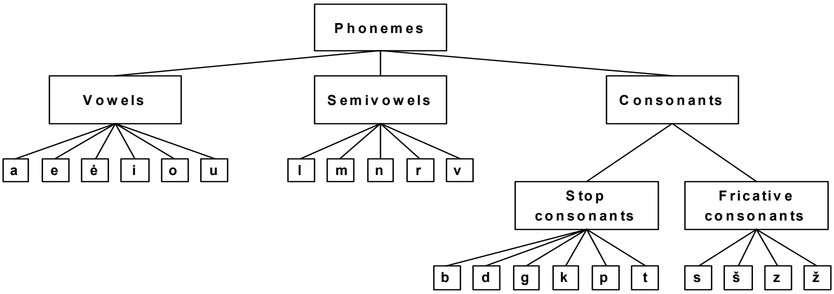


Figure 1. Hierarchical phoneme group's structure

3. Experiments

For these experiments we selected first eight phrases (number names and control words) of Lithuanian speech corpora LTDIGITS. We used utterances of 200 speakers (100 male, 100 female). The total number of words in the experiments was approximately 9600. Feature vectors contained 12 MFCC coefficients, energy and their first and second derivatives (delta and delta-delta coefficients). Then we used algorithm which will find MFCC coefficients, corresponding to particular phoneme from results of manual acoustic-phonetic labeling. Since duration of each phoneme is different and this means different number of frames, the following algorithm was realized to get template MFCC based vector for each phoneme: at first we take central (or steady-state part) of phoneme (it‘s duration about 50 ms) and then we calculate mean of each parameter in the MFCC vector per 8 frames. Later from right edge of phoneme to it‘s center we take MFCC coefficients and also calculate means of 8 frames. In this way we get MFCC template of the right context of phoneme. Similarly we got template of left context of phoneme. For these experiments we used only templates obtained from steady-state part of phoneme.

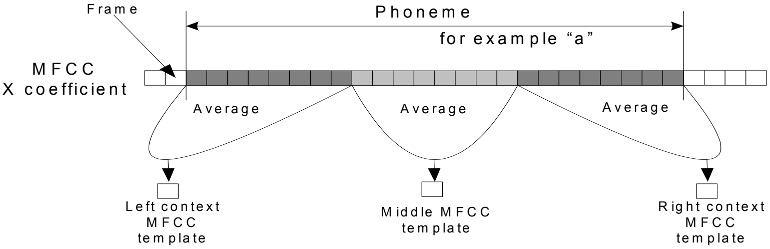


Figure 2. Phoneme template calculation scheme

Experiments were carried in two stages. At first it was realized phoneme recognition system without hierarchical structure. 21 phoneme templates were prepared and during recognition each speech signal part has been compared with all phoneme templates. There were totally 5082 phonemes in the recognition stage (from utterances of 20 speakers). At the second stage hierarchical method was used: phoneme has been recognized to one of the phoneme group and only after that phoneme has been recognized to exact class.

4. Main results and discussion

Analyzing phoneme class recognition results (Table 1) we could observe that using hierarchical phoneme recognition structure classification into the vowels, semivowels, plosives and fricatives general accuracy was improved by 5 percent (from 79,6 percent up to 84,5 percent). Looking at the recognition results of each class it could be seen that most significant improvement (14 percent) was achieved on vowels. Unfortunately recognition accuracy of semivowels and plosives worsened.

Looking at the vowel classification results (Table 2) we could observe that biggest increase in accuracy was on vowels ***I*** and ***E.*** Accuracy of ***O*** and ***Ė*** recognition remained the same while error rate of vowel ***U*** increased by 4 percent. In latter case degradation of performance was caused by confusion with the semivowel ***L*** and plosive ***P***.

The worst results using structural approach were observed in the semivowels group. The main reason of this was deterioration of phoneme ***R*** recognition. It could be speculated that this was caused by the fact that when deriving covariance matrix for this group majority of used phonemes where ***M,N*** while the amount of ***V***, ***L*** and vibrants was smaller. So we think that in the future semivowels (or liquids) should not be joined to the single class but instead subdivided into classes of nasals, laterals and vibrant liquids. Experimental results showed that articulation of these sounds has more differences than similarities.

Accuracy of fricatives classification increased slightly also (from 77,7 percent up to 80,9 percent). Analysis of results shows that voiced consonants ***Z, Ž*** most frequently were confused with the voiced plosives ***D, G***, while unvoiced consonant ***S*** with unvoiced plosives ***K, P, T.***

Table 1. Phoneme group discrimination results with and without hierarchical structure method

	Vowels	**Semivowels**	**Plosives**	**Fricatives**
number of phonemes	2301	886	1276	619
	Recognition without hierarchical structure			
as vowel	**74.1**	11.5	1.0	1.5
as semivowel	21.5	**74.8**	5.3	12.9
as plosive	4.4	13.5	**93.7**	7.9
as fricative	0	0.1	0	**77.7**
	Recognition using hierarchical structure			
as vowel	**87.9**	13.5	1.6	0.5
as semivowel	4.4	**69.0**	7.7	0.2
as plosive	7.6	17.5	**90.7**	18.4
as fricative	0	0	0	**80.9**

The use of hierarchical classification allowed to obtain faster performance of speech recognition system by approximately 50% (two computers with different speech and resources were used in evaluation).

Table 2. Discrimination results of vowels with hierarchical recognition approach and without it.

	A	**E**	**Ė**	**I**	**O**	**U**
number of phonemes	442	437	38	925	134	325
	Recognition without hierarchical structure					
correctly	**76.0**	**45.3**	**13.2**	**55.9**	**67.2**	**67.7**
as vowel	**86.0**	**71.9**	**94.7**	**65.6**	**94.0**	**74.5**
As semivowel	11.0	25.9	5.3	29.0	5.3	17.5
as plosive	3.4	2.3	0	5.3	0.7	8.0
as fricative	0	0	0	0.1	0	0
	Recognition using hierarchical structure					
correctly	**82.6**	**58.8**	**13.2**	**70.4**	**67.2**	**63.4**
as vowel	**93.7**	**91.5**	**100**	**86.8**	**97.8**	**72.9**
as semivowel	2.0	2.5	0	5.3	0	9.8
as plosive	4.3	5.9	0	7.9	2.2	16.9
as fricative	0	0	0	0	0	0.3

5. Conclusions

Concluding the results of our investigation it could be noticed that after integration of hierarchical phoneme structure overall performance increased by 2 percent (from 57,6 percent up to 59,6 percent). Such moderate improvement may be caused by the inability of linear discriminant approach (Fisher classifier) with MFCC derived feature vectors to capture all properties of phoneme groups. We plan to implement discrimination of phoneme groups using additional acoustic- phonetic features in the future. Modifications of MFCC feature vectors could be introduced also.

Phoneme classes recognition accuracy increased by 5 percent. Hierarchical classification was particularly efficient in the class of vowels. At the same time recognition performance slightly deteriorated in the classes of semivowels and plosives. These results could be caused by the unequal number of different phonemes used to construct templates of the phoneme classes. Consequently this was caused by the unequal number of different phonemes available in speech corpora. In the future we plan to evaluate phoneme frequencies in the corpora. Performed experiments permit to conclude that MFCC coefficients has some discriminational information about voiced and unvoiced sounds so we suppose to adjust hierarchical model structure by incorporating classes of voiced and unvoiced sounds additionally.

References

Girdenis, Aleksas 1995. Theoretical foundations of phonology. Vilnius: Petro ofsetas (in Lithuanian)

Koizumi, Takuya, Mori, Mikio, Taniguchi, Suji, Maruya. 1996. Reccurent neural networks for phoneme recognition. In *Proceedings Fourth International Conference ICSLP'96*, Philadelphia, vol. 1, pp. 326 –329

Lippmann, Richard. Recognition by Humans and Machines: Miles to Go Before We Sleep. In *Speech Communication*, vol. 18, April 1996, pp. 1-15.

Rudžionis, Algimantas, Rudžionis Vytautas 1999. Phoneme recognition in fixed context using regularized discriminant analysis. In. *Proceedings 6th European conference on Speech Communication and Technology, Eurospeech'99*, Budapest, Hungary, vol. 3, pp. 2745-2748

Šilingas, Darius, Laurinčiukaitė, Sigita, Telksnys, Laimutis 2004. Towards acoustic modeling of Lithuanian speech. In: *Proceedings of SPECOM, 2004* St.Petersburgh, Russia.

VYTAUTAS RUDZIONIS is senior researcher and associated professor in the Kaunas faculty of Vilnius University, Kaunas, Lithuania. He received his doctoral degree at the Kaunas University of technology. His primary research interests are in the speech recognition, particularly phoneme based speech recognition, problems. Other interests are in the signal processing, speech compression and related areas. Particular interest author pays to the speech technology implementation and design of speech technology enabled systems. He teaches several courses (both graduate and undergraduate) at Kaunas faculty of Vilnius University, between them courses on the basics of speech technology (graduate) and basics of artificial intelligence (undergraduate). E-mail: vyrud@mmlab.ktu.lt

KESTUTIS DRIAUNYS is PH. D. student in the Kaunas faculty of Vilnius University. He is preparing doctoral dissertation which is focused on speech recognition improvement through improved phonemic recognition. His interests cover speech recognition, signal processing, neural networks. Kestutis Driaunys published 2 articles in scientific journals and 4 in conference proceedings. Besides research he teaches different courses related with computer science. E-mail: kestutis.driaunys@vukhf.lt

PRANAS ZVINYS is Deputy Dean at Vilnius University Kaunas Faculty of Humanities and associated professor at department of Informatics. He graduated radioelectronics engineer studies at Kaunas Politechnical Institute (now Kaunas University of Technology). He received doctoral degree at the Kaunas Politechnical Institute. His research interests are in the continuous speech signal segmentation and labeling, speech recognition. He teaches some courses at Vilnius University Kaunas faculty of Humanities: basics of algorithmization, information technologies and others. E-mail: pranas.zvinys@vukhf.lt

USING SENSE CATEGORIES AND WSD SYSTEM SCORES TO GENERATE APPLICATION-SPECIFIC SENSE INVENTORIES

Harri M. T. Saarikoski
Helsinki University (Finland)

Abstract

Sense inventory generation is not only a task for lexical resource designers but also a pre-stage for word sense disambiguation (WSD) and NLP applications requiring disambiguation. This paper presents a novel approach to solve the laborious generation of application-specific sense inventories for e.g. machine translation and information retrieval applications. Our method is to search for correlations between translations (EuroWordNet) and two types of sense distinctions: (1) Human: sense categories from WordNet database (lexicographer files) and SENSEVAL evaluations (coarse grain sense inventory) and (2) Automatic: word sense disambiguation system scores from SENSEVAL evaluations. Systematic and significant correlations would enable us to use one large 'NLP sense inventory' (based on WordNet) incorporated with rules to generate the different sub-inventories. Our test shows that correlations between desired translation grain and both sense categories well exceed the random untrained baseline, while WSD system scores only manifest a complex pattern that does not systematically correlate with translation clusters.

Keywords: word sense disambiguation, SENSEVAL, NLP application, machine translation, information retrieval, WordNet, lexicographer files, lexical tuning

1. Introduction

Large-scale, general sense inventories (like WordNet) often fail to supply the appropriate grain for a given NLP application type (information retrieval, machine translation, and so on) (see Resnik et al. 1997). Automating the huge undertaking of sense inventory generation would be a great advance not only for sense inventory design but also to evolution of WSD and NLP applications.

Not much work has been done in this area. Mihalcea et al. (2001) designed a method that successfully generates a coarse-grain inventory out of WordNet by collapsing word senses that are most similar in their synsets and related synsets. Chugur et al. (2002) used similarity of cross-lingual translations in bilingual aligned corpora to generate sense inventories for machine translation.

Our own approach is an adaptation of the two: we look for correlations (of sense categories in WordNet and SENSEVAL-2 evaluation) in the word's translation equivalents in dictionaries of multiple languages. A sample set of 18 nouns from SENSEVAL-2 English lexical sample was tested.

2. Test

In this section, we describe the features used to find any correlations.

2.1. Translation clusters

As translation languages we tested Swahili (translations obtained from Kamusi ya kiswahili sanifu 1981 and TUKI 2001) and Spanish (EuroWordNet). We compared the translations to the sense categories and system scores (see 2.2 and 2.3). If the translations for any two senses were the same, those senses were assimilated into a 'translation cluster', which constituted the application-specific sense inventory. Senses with no translation available for either language were omitted.

2.2. Sense categories

We used WordNet lexicographer files (also known as supersenses) and SENSEVAL-2 coarse-grain sense inventory as sense categories. Lexicographer files contain around 25 categories (e.g. Artifact, Cognition) both for nouns and verbs and cover the whole of WordNet. The coarse-grain inventory, on the other hand, is a resource manually designed for the SENSEVAL-2 English lexical sample task, where a total of 73 words were grouped based on syntactic usage and logical division.

2.3. WSD system scores

We also calculated the average precision of all SENSEVAL-2 English lexical sample systems (supervised, unsupervised and baseline) that attempted to disambiguate the senses of the 18-word sample (see SENSEVAL, Edmonds et al. 2002). Senses for which there were less than 5 test instances were omitted.

3. Results

Below we describe the features (as listed in chapter 2) and the correlations found.

Table 1. Features for three most frequent senses of mouth[n]

Sense and precision	Description	Swahili	Spanish	Coarse	Lexfile
Mouth10800 (*0.476*)	the opening through which food is taken	Mdomo	Boca	1	Body
Mouth10801 (*0.492*)	externally visible part of the oral cavity	Mdomo Kinywa	Boca	1	Body
Mouth11700 (*0.575*)	the point where a stream issues into a larger body of water	Mlango	Desembocadura	2	Object

In Table 1, we used shadings (grey) to indicate correlation over and within categories. Both languages (Swahili and Spanish) as well as both coarse clusters (Coarse) and lexicographer files (Lexfile) clustered the senses in the same groups, which means that the desired translation clusters could be obtained from the sense categories. WSD system scores (Sense and precision), on the other hand, did not appear to correlate in any unambiguous way with translation clusters in the two target languages over the word sample.

Table 2. Precision percentages for sense inventory generation systems

Number of senses	**Coarse**	**Lexfile**	**Random**
2	33	**60**	50
3	**80**	60	25
4	**33**	**33**	12
5 and 6	**25**	**25**	6
Average	**61**	50	22

In Table 2, Coarse and Lexfile columns refer to systems that use coarse-grain sense inventory and lexicographer files, respectively. Their results are shown here against random baseline (untrained). We can see that our methods exceed the baseline quite distinctly, regardless of the number of senses. The correlation seems to depend on the target word. For example, the following words showed high correlation (i.e. success of the method): *lady* (2 senses: generic woman or fine lady), *mouth* (3 senses) and *authority* (4). On the other hand, correlations were low and unusable with words like *feeling* (2), *sense* (4) and *channel* (5).

4. Conclusion

We found strong evidence that the formation of translation clusters correlates with both WordNet lexicographer files and SENSEVAL coarse-grain sense inventory. Semantic sense categories can therefore be used to predict same/different translation equivalents in a given target language. Reliability far beyond the baseline can be expected. Although translations of word senses have evolved separately and under different laws than semantic categorizations of those senses, they also seem to have a common ground. Despite the preliminary nature and low statistical weight of the results, the method (even only slightly trained) shows promise in making a low-cost and full-coverage mapping between sense categories and translations.

Acknowledgment

I owe the English-Swahili translations to Ms. Wanjiku Nganga of Helsinki University.

References

Chugur, I., Gonzalo, J. and Verdejo, F. 2002. Polysemy and sense proximity in the Senseval-2 test suite. In: *Proceedings of the ACL SIGLEX workshop on WSD (SENSEVAL): recent successes and future directions*.

Edmonds, P.; Kilgarriff, A. 2002. Introduction to the Special Issue on evaluating word sense disambiguation programs. In: *Journal of Natural Language Engineering 8(4)*.

EuroWordNet. Online user interface at http://nipadio.lsi.upc.es/cgi-bin/wei/public/wei.consult.perl.

Kamusi ya kiswahili sanifu: Standard Swahili Dictionary 1981. Nairobi, Kenya: Oxford University Press – East Africa.

Mihalcea, R.; Moldovan, D. 2001. EZ.WordNet: Principles for automatic generation of a coarse grained WordNet. In: *Proceedings of Florida Artificial Intelligence Research Society Conference (FLAIRS 2001).*

Resnik, P.; Yarowsky. D. 1997. A perspective on word sense disambiguation methods and their evaluation. In: *Proceedings of ACL SIGLEX workshop on tagging text with lexical semantics: why, what and how?*

Saarikoski, H. 2004. Using SENSEVAL system scores to optimize lexical resources for WSD and knowledge acquisition. In: *Proceedings of Iberamia, IX Ibero-American Workshop on Artificial Intelligence.*

SENSEVAL. WSD system evaluation data obtainable from http://www.senseval.org/.

TUKI: English-Swahili dictionary 2001. University of Dar es Salaam, Tanzania: Institute of Kiswahili Research.

WordNet 2.0. Online user interface at www.cogsci.princeton.edu/cgi-bin/webwn.

HARRI SAARIKOSKI is a doctorate student of Department of General Linguistics (Language Technology) of Helsinki University and AAC Global Ltd. His studies focus on exploring the state-of-art word sense disambiguation methodologies for latent potential, primarily with the help of error analysis of existing systems and performance predictions using large-scale lexical resources. E-mail: harri.saarikoski@helsinki.fi

BUILDING COMPACT N-GRAM LANGUAGE MODELS INCREMENTALLY

Vesa Siivola
Neural Networks Research Centre, Helsinki University of Technology, Finland

Abstract

In traditional n-gram language modeling, we collect the statistics for all n-grams observed in the training set up to a certain order. The model can then be pruned down to a more compact size with some loss in modeling accuracy. One of the more principled methods for pruning the model is the entropy-based pruning proposed by Stolcke (1998). In this paper, we present an algorithm for incrementally constructing an n-gram model. During the model construction, our method uses less memory than the pruning-based algorithms, since we never have to handle the full unpruned model. When carefully implemented, the algorithm achieves a reasonable speed. We compare our models to the entropy-pruned models in both cross-entropy and speech recognition experiments in Finnish. The entropy experiments show that neither of the methods is optimal and that the entropy-based pruning is quite sensitive to the choice of the initial model. The proposed method seems better suitable for creating complex models. Nevertheless, even the small models created by our method perform along with the best of the small entropy-pruned models in speech recognition experiments. The more complex models created by the proposed method outperform the corresponding entropy-pruned models in our experiments.

Keywords: variable length n-grams, speech recognition, sub-word units, language model pruning

1. Introduction

The most common way of modeling language for speech recognition is to build an n-gram model. Traditionally, all n-gram counts up to a certain order n are collected and smoothed probability estimates for words are based on these counts. There exist several heuristic methods for pruning the n-gram model to a smaller size. One can for example set cut-off values, so that the n-grams that have occurred less than m times are not used for constructing the model. A more principled approach is presented by Stolcke (1998), where the n-grams, which reduce the training set likelihood the least are pruned from the model. The algorithm seems to be effective in compressing the models with reasonable reductions in the modeling accuracy.

In this paper, an incremental method for building n-gram models is presented. We start adding new n-grams to the model until we reach the desired complexity. When deciding if a new n-gram should be added, we weight the training set likelihood increase against the resulting growth in model complexity. The approach is based on the Minimum Description Length principle (Rissanen 1989). The algorithm presented here has some nice properties: we do not need to decide the highest possible order of an n-gram. The construction of the model takes less memory than with the entropy based pruning algorithm, since we are not pruning an existing large model to a smaller size, but extending an existing small model to a bigger size. On the downside, the algorithm has to be carefully implemented to make it reasonably fast.

All experiments are conducted on Finnish data. We have found that using *morphs*, that is statistically learned morpheme like units (Creutz and Lagus 2002) as a basis for an n-gram model is more effective, than using a word-based model. The first experiments (Siivola et al. 2003) were confirmed by later experiments with a wider variety of models and the morphs were found to consistently outperform other units. Consequently, we will be using the morph-based n-gram models also in the experiments of this paper. We compare the proposed model to an entropy-pruned model in both cross-entropy and speech recognition experiments.

2. Description of the method

The algorithm is formulated loosely based on the Minimum Description Length criterion (Rissanen 1989), where the object is to send given data with as few bits as possible. The more structure is contained in the data, the more useful it is to send a detailed model of the data, since the actual data can be then described with fewer bits. The coding length of the data is thus the sum of the model code length and the data log likelihood.

2.1. Data likelihood

Assume that we have an existing model $\mathcal{M}_o$ and we are trying to add n-grams of order n into the model. We start by drawing a *prefix gram*, that is an $(n-1)$-gram g_{n-1} from some distribution. Next, we try adding all observed n-grams g_n starting with the prefix g_{n-1} to the model to create a new model $\mathcal{M}_n$. The change of the log likelihood $L_{\mathcal{M}}$ of the training data T between the models is

$$\Lambda(\mathcal{M}_n, \mathcal{M}_o) = L_{\mathcal{M}_n}(T) - L_{\mathcal{M}_o}(T) \tag{1}$$

Adding the n-grams g_n increases the complexity of the model. We want to weight the gain in likelihood against the increase in the model complexity.

2.2. Model coding length

We are actually only interested in the change of the model complexity. Thus, if we assume our vocabulary to be constant, we need not to think about coding it. For each n-gram g_n, we need to store the probability of the n-gram. The interpolation (or back-off) coefficient is common to all n-grams g_n starting with the same prefix g_{n-1}. As n-gram models tend to be sparse, they can be efficiently stored in a tree structure (Whittaker and Raj 2001). We can claim, that adding n-gram of any order into the tree demans an equal increase in model size, if we make the approximation that all n-grams are prefixes to other n-grams. This means that all n-grams need to store an interpolation coefficient correspondig to the n-grams they are the prefix to. Also, all n-grams also need to store what Whittaker and Raj call *child node index*, that is the range of child nodes of a particular n-gram prefix. Accordingly, if the n-gram prefix needed for storing interpolation coefficient or child node index is not in the model, we need to add the corresponding n-gram.

The approximated cost Ω for updating the model is

$$\Omega(\mathcal{M}_n, \mathcal{M}_o) = n \cdot (2\log_2(W) + 2\theta) = nC, \tag{2}$$

where W is the size of the lexicon, n is the number of new n-grams in model $\mathcal{M}_n$, the cost of $2\log_2(W)$ comes from storing the word and child node indices. The cost 2θ comes from storing the log probability and the interpolation coefficient with the precision of θ bits.

2.3. N-gram model construction

The n-gram model is constructed by sampling the prefixes g_{n-1} and adding all n-grams g_n starting with the prefix, if the change Δ in data coding length Ψ is negative.

$$\Delta = \Psi(\mathcal{M}_n) - \Psi(\mathcal{M}_o) = \Omega(\mathcal{M}_n, \mathcal{M}_o) - \alpha\Lambda(\mathcal{M}_n, \mathcal{M}_o) \tag{3}$$

We have added the coefficient α to scale the relative importance of the training set data. We are not trying to encode a certain data set, but we are trying to build an optimal n-gram model of certain complexity. With α, we can control the size of the resulting model.

There is also a fixed threshold, which the improvement of the data log likelihood $\Lambda(\mathcal{M}_n, \mathcal{M}_o)$ has to exceed, before the new n-grams are even considered for inclusion to the model. Originally this was to speed up the model construction, but it seems that the resulting models are also somewhat better.

For sampling the prefixes we used a simple greedy search. We go through the existing model, order by order, n-gram by n-gram and use these n-grams as the prefix grams. For the n-gram probability estimate, we have used modified Kneser-Ney smoothing (Chen and Goodman 1999). Instead of using estimates for optimal discounts, we decided use Powell search (Press et al. 1997) to find the optimal parameter values, since the n-gram distribution of the model was quite different from a model, where all n-grams of a given order from the training set are present. The discount parameters are re-estimated each time when 10000 new prefixes have been added to a new n-gram order.

2.4. Morphs

For splitting words into morpheme-like units, we use slightly modified version of the algorithm presented by Creutz and Lagus (2002). The word list given to the algorithm was filtered so, that all words with frequency less than 3 were removed from the list. Word counts were ignored, all words were assumed to have occurred once. This resulted in a lexicon of 26 000 morphs.

2.5. Details of the implementation

It is important to consider the implementation of the algorithm carefully. A naive implementation will be too slow for any practical use. In all places of the algorithm, where there is calculation of differences, we only modify and recalculate the parameters, which affect the difference.

When we have sampled a prefix, we have to find the corresponding n-gram counts from the training data. For effective search, we have a word table, where each entry contains an ordered list of locations, where the word has been seen in the training set. We use a slightly modified binary search, starting from the rarest word of the n-gram to find all the occurrences of the n-gram.

We initialized our model to unigram model. It would be possible to start the model construction from 0-grams instead of unigrams. This is maybe a theoretically nicer solution, but in practice we suspect that all words will have at least their unigram probabilities estimated anyway.

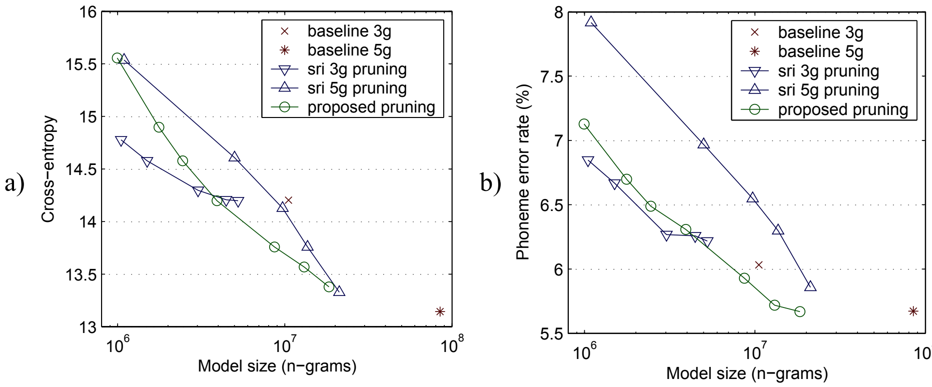


Figure 1: Experimental results. The model sizes are expressed on a logarithmic scale. a) Cross-entropies against the number of the n-grams in the model. The measured points on each curve correspond to different pruning or growing parameter values. b) Phoneme errors and model sizes. Corresponding word error rates range from 25.5% to 39.6%.

3. Experiments

3.1. Data

We used some data from the Finnish Language Bank (CSC 2004) augmented by an almost equal amount of short newswires, resulting in corpus of 36M words (100M morphs). 50k words were set aside as a test set.

The audio data was 5 hours of short news stories read by one female reader. 3.5 hours were used for training, the LM scaling factor was set based on a development set of 33 minutes and finally 49 minutes of the material were left as the test set.

3.2. Cross-entropy

We trained an unpruned baseline 3-gram and 5-gram model from the data to serve as reference models. We used the SRILM–toolkit (Stolcke 2002) to train the entropy-pruned models and compared these against our models. Both the proposed and entropy based pruning method were run with different parameter values for pruning or growing the model. For testing the models, we calculated the cross-entropy of the model $\mathcal{M}$ and the test set text T:

$$H_M(T) = -\frac{1}{W_T} \log_2 P(T|\mathcal{M}) \tag{4}$$

where W_T is the number of the words in the test set.

The cross-entropy is directly related to perplexity, but seems to reflect the changes in word error rates better, which is why we used it. The results for the models are plotted in Figure 1a. From Figure 1a we see that the proposed model is consistently better than the pruned 5-gram model from the SRILM toolkit. The pruned 3-gram model from the SRILM toolkit is more effective in creating small models than the proposed model. It seems that both the SRILM pruning and the proposed algorithms are suboptimal, since the results should be at least as good as from any pruned 3-gram model.

In Figure 2 we have plotted the distribution of n-grams in pruned SRILM models and in the proposed models. We see that the n-gram distribution in our model is more weighted towards the lower order n-grams.

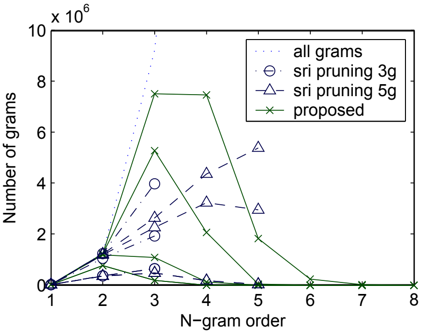


Figure 2: N-gram distributions of pruned SRILM models and the proposed models. The plot shows the number of n-grams of each order in a model. The points belonging to the same model are connected with a line.

3.3. Speech recognition system

Our acoustic features were 12 Mel-Cepstral coefficients and power. The feature vector was concatenated with corresponding first order delta features. The acoustic models were monophone HMMs with Gaussian Mixture Models. The acoustic models had explicit duration modeling, the post-processor approach presented by Pylkkönen and Kurimo (2004). Our decoder is a so-called stack decoder (Hirsimäki and Kurimo 2004).

3.4. Speech recognition experiments

The speech recognition experiments were run on the same models as the cross-entropy experiments. The phoneme error rate of the models has been shown in Figure 1b. The recognition speeds ranged from 1.5 to 3 times real time on an AMD Opteron 248 machine. Tightening the pruning to faster than real time recognition leads to a very similar figure, with phoneme error rates ranging from 6.2% to 8.4%.

The proposed model seems to do relatively better in the speech recognition experiments than in the cross-entropy experiments. This is probably because the n-gram distribution of the proposed model is more weighted towards the lower order n-grams. This way, the speech recognition errors affect a smaller number of utilized language model histories. It seems likely, that the decoder prunings also play some role.

4. Discussion and conclusions

We presented an incremental method for building n-gram language models. The method seems well suitable for building all but the smallest models. The method does not use a fixed n for building n-gram statistics, instead it incrementally expands the model. The model uses less memory when creating the model than the comparable pruning methods. The experiments show, that the proposed method robustly gets similar results as the existing entropy-based pruning method (Stolcke 1998), where a good choice of the initial n-gram order is required.

It seems that both the proposed and entropy based pruning method are suboptimal. In theory, an optimal pruning started from a 5-gram model should always be better than or equal to an optimal pruning started from a trigram model. When creating small models, the entropy based pruning from trigrams gives better results than either the proposed method or entropy based pruning from 5-grams.

One possible reason for the suboptimal behavior is that both methods use greedy search for finding the best model. The search is not guaranteed to find the optimal model. Also, neither of the models takes into account that the lower order n-grams will probably be proportionally more used in new data than the higher order n-grams. In our model we made some crude approximations when estimating the cost of adding new n-grams to the model. More accurate modeling of the cost of inserting an n-gram to the model would penalize the higher order n-grams somewhat and possibly lead to improved models.

The models should be further tested with a wide range of different training set sizes and word error rates to get a more accurate view how the models perform compared to each other in more varying circumstances. We chose to use morphs as our base modeling units, but the presented method should also work on word-based models. Experiments should be run on languages where word-based models work better, such as English.

5. Acknowledgements

This work was funded by Finnish National Technology Agency (TEKES). The author thanks Mathias Creutz for the discussion leading to the development of this model and our speech group for helping with the speech recognition experiments. The Finnish news agency (STT) and the Finnish IT center for science (CSC) are thanked for the text data. Inger Ekman from the University of Tampere, Department of Information Studies, is thanked for providing the audio data.

References

CSC 2004. Collection of Finnish text documents from years 1990–2000. Finnish IT center for science (CSC).

Chen, Stanley F.; Goodman, Joshua 1999. An empirical study of smoothing techniques for language modeling. In: *Computer Speech and Language* **13(4)**, 359–393

Creutz, Mathias; Lagus, Krista 2002. Unsupervised discovery of morphemes. In: *Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02*. 21–30

Hirsimäki, Teemu; Kurimo, Mikko 2004. Decoder issues in unlimited Finnish speech recognition. In: *Proceedings of the 6th Nordic Signal Processing Symposium (Norsig)*. 320–323

Press, William; Teukolsky, Saul; Vetterling, William; Flannery, Brian (eds.) 1997. Numerical recipes in C. Cambridge University Press

Pylkkönen, Janne; Kurimo, Mikko 2004. Using phone durations in Finnish large vocabulary continuous speech recognition. In: *Proc. Norsig 2004*. 324–326

Rissanen, Jorma 1989. Stochastic complexity in statistical inquiry theory. World Scientific Publishing Co., Inc.

Siivola, Vesa; Hirsimäki, Teemu; Creutz, Mathias; Kurimo, Mikko 2003. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In: *Proc. Eurospeech 2003*. 2293–2296

Stolcke, Andreas 1998. Entropy-based pruning of backoff language models. In: *Proc. DARPA Broadcast News Transcription and Understanding Workshop*. 270–274

Stolcke, Andreas 2002. SRILM – an extensible language modeling toolkit. In: *Proc. ICSLP 2002*. 901–904. `http://www.speech.sri.com/projects/srilm/`

Whittaker, E.W.D.; Raj, B. 2001. Quantization-based language model compression. In: *Proc. Eurospeech 2001*. 33–36

VESA SIIVOLA is a graduate student (M.Sc.) working as a researcher in Neural Networks Research Centre, Helsinki University of Technology. E-mail: Vesa.Siivola@hut.fi .

A SHORT INTRODUCTION TO UNIFICATIONAL COMBINATORY CATEGORIAL GRAMMAR

Maarika Traat
The University of Edinburgh, UK

Abstract

This paper describes Unificational Combinatory Categorial Grammar (UCCG): a grammar formalism that combines the insights from Combinatory Categorial Grammar (CCG) (Steedman 2000) with feature structure unification. UCCG is a new member of the family of categorial grammars. It is particularly closely related to two other members of the family – CCG and Unification Categorial Grammar (UCG) (Calder et al. 1988; Zeevat 1988). Similarly to UCG, UCCG has a very close relationship between syntax and semantics achieved by unification, but with the added flexibility provided by the use of CCG combinatory rules. UCCG builds upon UCG's representation of linguistic expressions as feature structures, but in addition, UCCG uses the directional slash representation of CCG. For semantics, UCCG adopts the DRT representation.

Keywords: grammar formalisms, syntax-semantics interface, DRT, feature structures, signs, categories, combinatory rules

1. Introduction

Unificational Combinatory Categorial Grammar (UCCG) is a member of the family of categorial grammars. It is particularly closely related to Combinatory Categorial Grammar (CCG) and Unificational Categorial Grammar (UCG). The special beauty of UCCG lies in the fact that employing unification as the main machinery in combining grammatical categories it is easy to implement in a computer language like Prolog, but in spite of its computer friendliness, its representations are very easily readable by a human. UCCG combines several by now widely accepted approaches representing linguistical expressions as feature structures familiar from HPSG, and using standard Discourse Representation Theory (DRT) (Kamp and Reyle 1993) semantics for the representation of meaning. Although, UCCG was particularly created for handling intonational features ((Traat and Bos 2004)), the present paper describes the general framework omitting the intonational part. Before taking a closer look at UCCG we make some remarks about its ancestors.

Categorial Grammars (CG) (Wood 2000) are lexicalised theories of grammar. The name of the grammar family comes from the notion of "category", i.e. the functional type that is associated with each entry in the lexicon which determines the ability of a lexical item to combine with other lexical items. CGs also have a set of rules defining the syntacto-semantic operations that can be performed on the categories.

Combinatory Categorial Grammar (CCG) (Steedman 2000) is a generalisation of CG. While the pure CG only involves functional application rules for combining categories, CCG introduces several additional combinatory rules for both syntactic and semantic combination – forward and backward composition, crossed composition and substitution rules. For building semantic representation, CCG uses the lambda calculus, although unification has been proposed as well Steedman (1990).

Unification Categorial Grammar (UCG) (Calder et al. 1988) uses Head-Driven Phrase Structure Grammar (HPSG) type of feature structures, called signs, to represent the categories of lexical items. The directionality of the attributes of a functor category is marked by the features *pre* and *post* on its attributes rather than by the directionality of the slashes as done in CCG. For semantic representation UCG uses Indexed Language (InL).

2. Unificational Combinatory Categorial Grammar

UCCG aims to marry the best parts of CCG and UCG, whilst also making its own additions. From CCG it inherits the slash notation – both forward and backward slashes, and the combinatory rules. In contrast to UCG, which only uses forward and backward application as the operations of combining categories, UCCG also uses all the varieties of composition present in CCG, as well as type-raising. However, similarly to UCG, UCCG uses feature structures to represent the linguistic data, and both semantics and syntax are built up simultaneously via unification. In contrast with its two close relatives UCCG uses standard DRT semantics. The traditional version of DRT was chosen as the semantic representation because it is widely known and accepted as being well suited for computational language applications.

As far as grammatical categories are concerned, UCCG slightly differs from both CCG and UCG by its noun phrases always being complex categories. This requirement is due to the need to be able to control the quantifier scope of determiners. On the other hand intransitive verb phrases which are complex categories in CCG and UCG are collapsed into a basic category in UCCG.

2.1. UCCG Signs and Categories

In its linguistic description UCCG makes use of feature structures called signs. There are two types of signs – basic and complex. A UCCG basic sign is a list of attributes or features describing the syntactic and semantic characteristics of a lexical expression. A sign can have a varied number of features, depending on the syntactic category of the lexical expression the sign is characterising. There are three obligatory features any sign must have: PHO (phonological form), CAT (syntactic category), and DRS (semantic representation). A sign can also have the following features: AGR to mark the inflectional characteristics of verbs (e.g. finite or non-finite); VAR for a variable standing for discourse referents which plays an important role in unification, and creates a direct link between syntax and semantics; SIT for neo-davidsonian event variables. Depending of the needs of a specific application and language for which a UCCG grammar is constructed, many more features could be introduced in each basic sign.

There are three kinds of basic signs in UCCG, corresponding to the basic categories of UCCG: those with CAT feature sentence (s), noun (n) and verb phrase (vp) (see Ex. 2, 3, 4). Besides basic signs there are complex signs in UCCG. These are formed from basic signs by the use of CCG style slashes. The process of forming complex signs is recursive: complex signs can be combined to form even more complex signs (see Ex. 1).

More formally:

- *If X and Y are signs then X/Y is a complex sign.*
- *If X and Y are signs X\Y is a complex sign.*
- *All basic and complex signs are signs.*

Some more terminology (see also Ex. 1): X is the **result part** of a sign X/Y or X\Y; Y is the **argument part** of a sign X/Y or X\Y; Z is the **result head** of a result part Z/W or Z\W, where Z is the leftmost basic sub-sign in a complex sign.

(1)
←———— **result part** ————→ ←———— **argument part** ————→
←—— **result head** ——→

$$\left(\begin{bmatrix}\text{PHO:} & \text{gives+W+W1}\\ \text{CAT:} & \text{vp}\\ \text{AGR:} & \text{fin}\\ \text{VAR:} & \text{X}\\ \text{SIT:} & \text{E}\\ \text{DRS:} & \text{D}\end{bmatrix} / \left(\begin{bmatrix}\text{PHO:} & \text{W+W2}\\ \text{CAT:} & \text{s}\\ \text{DRS:} & \text{D}\end{bmatrix} / \begin{bmatrix}\text{PHO:} & \text{W2}\\ \text{CAT:} & \text{vp}\\ \text{AGR:} & \text{A}\\ \text{VAR:} & \text{Y}\\ \text{SIT:} & \text{E}\\ \text{DRS:} & \text{D1}\end{bmatrix}\right)\right) / \left(\begin{bmatrix}\text{PHO:} & \text{W1+W3}\\ \text{CAT:} & \text{s}\\ \text{DRS:} & \text{D1}\end{bmatrix} / \begin{bmatrix}\text{PHO:} & \text{W3}\\ \text{CAT:} & \text{vp}\\ \text{AGR:} & \text{A1}\\ \text{VAR:} & \text{Z}\\ \text{SIT:} & \text{E}\\ \text{DRS:} & \boxed{\begin{array}{l} \\ \hline \text{give(E)}\\ \text{actor(E,X)}\\ \text{beneficiary(E,Z)}\\ \text{patient(E,Y))}\end{array}}\end{bmatrix}\right)$$

Each sign has a syntactic category: for basic signs it corresponds to its CAT feature, for complex signs it is made up of the CAT features of all the component parts of the complex sign, separated by the slashes and brackets used in the complex sign. For example, the syntactic category of the complex sign in Example 1 is (vp/(s/vp))/(s/vp). Similarly to basic and complex signs we can speak about basic and complex categories. The three basic categories used in UCCG are thus *s*, *n* and *vp*, while all other categories are formed by combining these, using backward and forward slashes.

The special beauty of the variables in the UCCG feature structures lies in the fact that the same variables can be used at several different levels. For example, the variables standing for discourse referents serve as a link between syntax and semantics – the variable in the VAR feature in the feature structure fits into its corresponding slot in the DRS in the DRS feature. All the occurrences of the same variable get replaced simultaneously via unification when signs are combined.

The value of the VAR and the SIT features is always a variable, while other features can have a number of constant values. PHO feature holds the string value of the linguistic expression represented by the given feature structure. In basic signs the PHO feature is filled by lexical items, in complex signs it also contains variables, which get constant values when the result part sign is combined with its argument signs. The PHO feature in the result head of complex signs is of the form (. . . + W1) + *word* + (W2+. . .), where *word* is a lexical item, and W1 and W2 are variables that get values through unification in the categorial combination process. The item unifying with W1 precedes and that unifying with W2 follows the lexical item *word*. The exact number and order of the variables the PHO feature contains depends on the category of the given sign.

The AGR feature describes the inflectional properties of verbs, and is hence used in vp signs. It can take constant values *fin* (finite) or *non-fin* (non finite). Depending on the use of the formalism, the following features concerning various kinds of grammatical agreement could prove useful: PERSON, NUMBER and TENSE.

The value of the DRS (discourse representation structure) feature is either a variable, or holds a DRS corresponding to the semantics of the lexical item(s) characterised by the given sign. The DRS makes use of variables from the VAR and SIT features of the sign. UCCG DRSs use neo-davidsonian style event semantics (Landman 2000).

2.2. Basic Signs

There are the following three basic categories in UCCG: nouns (CAT:n), verb phrases (CAT:vp), and sentences(CAT:s). The lexical sign for nouns is very straightforward and its category directly corresponds to the CCG category n (see Ex. 2). In contrast with its two close relatives, UCCG introduces verb phrases among atomic signs (see Ex. 3). The basic UCCG vp category corresponds to the CCG category s\np. An example of a sentence sign can be seen in Example 4.

$$(2)\ \begin{bmatrix} \text{PHO:} & \text{child} \\ \text{CAT:} & \text{n} \\ \text{VAR:} & \text{X} \\ \text{DRS:} & \begin{array}{|l|}\hline \\ \hline \text{child(X)} \\ \hline\end{array}\end{bmatrix} \qquad (3)\ \begin{bmatrix} \text{PHO:} & \text{walks} \\ \text{CAT:} & \text{vp} \\ \text{AGR:} & \text{fin} \\ \text{VAR:} & \text{X} \\ \text{SIT:} & \text{E} \\ \text{DRS:} & \begin{array}{|l|}\hline \text{E} \\ \hline \text{walk(E)} \\ \text{actor(E,X)} \\ \hline\end{array}\end{bmatrix} \qquad (4)\ \begin{bmatrix} \text{PHO:} & \text{john + walks} \\ \text{CAT:} & \text{s} \\ \text{DRS:} & \begin{array}{|l|}\hline \text{X,E} \\ \hline \text{walk(E)} \\ \text{actor(E,X)} \\ \hline\end{array}\end{bmatrix}$$

2.3. Complex Signs and Combinatory Rules

Complex signs are constructed from basic signs and directional slashes. Similarly to CCG a forward slash in some sign X/Y means that the sign is looking for a sign of type Y on its right, while a backslash means that a sign of type W\Z is expecting a sign that can unify with its argument part Z on its left. Some of the complex signs are lexical, others are derived by combinatory rules.

2.3.1. Lexical Complex Signs

Determiners. The complex signs for determiners belong to the lexical complex signs. There is a major problem with assigning the determiners a category like *np/n* – it does not supply the means of specifying the scope of the determiner. Therefore, instead of the simple CCG *np* category, UCCG uses its type-raised counterpart *s/(s\np)*. In order to make the complex sign for noun phrases slightly more compact, we chose to represent *s\np* that stands for verb phrases as *vp* in UCCG. Thus, the category for determiners in UCCG is *(s/vp)/n*.

There are several signs for determiners, due to the different semantics that different determiners introduce into the noun phrase. The indefinite article takes the form seen in Example 5. The signs for determiners like *'every'* and *'no'* are very similar to the sign of the indefinite article, except for the semantics in the result head (see Ex. 6).

$$(5)\ \left(\begin{bmatrix} \text{PHO:} & \text{a+W1+W2} \\ \text{CAT:} & \text{s} \\ \text{DRS:} & (\begin{array}{|c|}\hline \text{X} \\ \hline \\ \hline\end{array};\text{D1};\text{D2}) \end{bmatrix} / \begin{bmatrix} \text{PHO:} & \text{W2} \\ \text{CAT:} & \text{vp} \\ \text{AGR:} & \text{fin} \\ \text{VAR:} & \text{X} \\ \text{SIT:} & \text{E} \\ \text{DRS:} & \text{D2} \end{bmatrix}\right) / \begin{bmatrix} \text{PHO:} & \text{W1} \\ \text{CAT:} & \text{n} \\ \text{VAR:} & \text{X} \\ \text{DRS:} & \text{D1} \end{bmatrix} \qquad (6)\ \text{a)} \begin{array}{|l|}\hline \\ \hline \begin{array}{|c|}\hline \text{X} \\ \hline \\ \hline\end{array};\text{D1}\Rightarrow\text{D2} \\ \hline\end{array} \quad \text{b)} \begin{array}{|l|}\hline \\ \hline \neg \begin{array}{|c|}\hline \text{X} \\ \hline \\ \hline\end{array};\text{D1};\text{D2} \\ \hline\end{array}$$

Noun phrases. Since most noun phrases (except e.g. proper nouns) are non-lexical we will discuss them in Section 2.3.2.

Verb Phrases. While the category of an verb phrase is just a basic sign – vp, then that of a transitive verb phrase is vp/(s/vp) and that of a ditransitive verb phrase is (vp/(s/vp))/(s/vp). A sign for a ditransitive verb phrase is illustrated above in Example 1.

Of course, there are many more lexical signs: those for adjectives, adverbs, prepositions, auxiliary verbs, etc.

2.3.2. Combinatory Rules and Combined Signs

When introducing complex categories and signs in Chapter 2.3.1 we often mentioned about some sign expecting a certain sign on its left or its right. But what happens if it finds it? The answer to this is given by combinateory rules, of which the most frequently used can be seen in Table 1. The rule table is to be interpreted in the following way: in the first column there is a combinatory rule, in the second the marking for it that will be used in derivations, and in the third the name of the rule. The variables *X*, *Y* and *Z* in the rules above stand for signs, which can be either basic or complex signs.

Table 1: Combinatory rules most frequently used in UCCG

$X/Y\ Y \Rightarrow X$	$\xrightarrow{\hspace{2em}}$ >	*Forward application*
$Y\ X\backslash Y \Rightarrow X$	< $\xleftarrow{\hspace{2em}}$	*Backward application*
$X/Y\ Y/Z \Rightarrow X/Z$	———*Comp>*	*Forward composition*
$Y\backslash Z\ X\backslash Y \Rightarrow X\backslash Z$	*<Comp*——-	*Backward composition*
$X \Rightarrow T/(T\backslash X)$ *or* $T\backslash(T/X)$	————*-T>*	*Type-raising*

Combined noun phrases. The simplest noun phrases are formed by combining a determiner and a noun by forward application. Example 7 demonstrates forming a noun phrase from the indefinite article '*a*' and a noun '*baby*'. The following is achieved by unification. The variable W1 in the PHO feature in the argument part of the determiner sign unifies with the value '*baby*' in the PHO feature in the noun sign, and through unification this value is also introduced as part of the value of the PHO feature in the result head of the determiner sign. The CAT n features in the argument part of the determiner sign and the noun sign unify, since they have the same constant value. Variable X in the VAR feature of the noun portion of the functor sign is unified with the variable Y in the argument sign. The variable S1 in the DRS feature in the argument part of the determiner sign unifies with the corresponding value in the DRS feature in the noun sign. Due to the use of the same variable the new value of the variable S1 is introduced in the semantics of the result head.

3. Conclusion and Future Work

This paper gave a short overview of Unificational Combinatory Categorial Grammar. It showed how UCCG has incorporated useful features from CCG and UCG, merging the result with standard DRT semantics. It explained the central notion of signs in UCCG, and the use of combinatory rules, illustrating the theory with examples of UCCG signs as well as derivations.

We have implemented a UCCG parser for a fragment of English. A future task would be to implement a limited domain UCCG generator that could be used in spoken dialogue systems to generate speech output with context appropriate intonation. Some other issues that we want to address are the three-fold relation between intonation, information structure and presupposition, and the possibility of automatically calculating presupposition while parsing prosodically annotated data.

(7)

$$\left\{\begin{array}{l}
\left(\begin{bmatrix}\text{PHO:} & \text{a+W1+W2}\\ \text{CAT:} & \text{s}\\ \text{DRS:} & \begin{array}{|c|}\hline \text{X}\\ \hline \text{D1;D2}\\ \hline\end{array}\end{bmatrix} / \begin{bmatrix}\text{PHO:} & \text{W2}\\ \text{CAT:} & \text{vp}\\ \text{AGR:} & \text{fin}\\ \text{VAR:} & \text{X}\\ \text{SIT:} & \text{E}\\ \text{DRS:} & \text{D2}\end{bmatrix}\right) / \begin{bmatrix}\text{PHO:} & \text{W1}\\ \text{CAT:} & \text{n}\\ \text{VAR:} & \text{X}\\ \text{DRS:} & \text{D1}\end{bmatrix} \begin{bmatrix}\text{PHO:} & \text{baby}\\ \text{CAT:} & \text{n}\\ \text{VAR:} & \text{Y}\\ \text{DRS:} & \begin{array}{|c|}\hline \\ \hline \text{baby(Y)}\\ \hline\end{array}\end{bmatrix}\\
\hline
\begin{bmatrix}\text{PHO:} & \text{a+baby+W2}\\ \text{CAT:} & \text{s}\\ \text{DRS:} & \begin{array}{|c|}\hline \text{X}\\ \hline \text{baby(X)}\\ \hline\end{array}\text{;D2}\end{bmatrix} / \begin{bmatrix}\text{PHO:} & \text{W2}\\ \text{CAT:} & \text{vp}\\ \text{AGR:} & \text{fin}\\ \text{VAR:} & \text{X}\\ \text{SIT:} & \text{E}\\ \text{DRS:} & \text{D2}\end{bmatrix}
\end{array}\right. >$$

References

Calder, Jonathan; Klein, Ewan; Zeevat, Henk 1988. Unification categorial grammar: A concise, extendable grammar for natural language processing. In: *Proceedings of the 12th International Conerence on Computational Linguistics*, Budapest

Kamp, Hans; Reyle, Uwe 1993. From Discourse to Logic. London: Kluwer Academic Publishers

Landman, Fred 2000. Events and Plurality. The Jerusalem Lectures: Vol. 76 of *Studies in Linguistics and Philosophy*. Dordrecht: Kluwer Academic Publishers

Steedman, Mark 1990. Gapping as constituent coordination. In: *Linguistics and Philosophy* 13

Steedman, Mark 2000. The Syntactic Process. Cambridge, Massachusetts: The MIT Press

Traat, Maarika; Bos, Johan 2004. Unificational combinatory categorial grammar: A formalism for parsing and generating prosodically annotated text. In: *Proceedings of COLING 2004*

Wood, Mary McGee 2000. Syntax in Categorial Grammar: An Introduction for Linguists. ESSLLI 2000, Birmingham, England. ESSLLI coursebook

Zeevat, Henk 1988. In: U.Reyle; C.Rohrer (eds.), *Natural Language Parsing and Linguistic Theories*, D.Reidel Publishing Company

MAARIKA TRAAT is a PhD student at the University of Edinburgh, UK. She is working under the supervision of Mark Steedman and Johan Bos. Her doctoral study focuses on developing a semantic representation with information structure compatible with first order logic, and embedding this semantics in a categorial grammar formalism. Her other current research interests are the syntax and semantics of English cleft constructions, and calculating presuppositions.

TREEBANK-BASED RESEARCH AND E-LEARNING OF ESTONIAN SYNTAX

Heli Uibo*, Eckhard Bick**
*University of Tartu (Estonia), **University of Southern Denmark

Abstract

The creation of syntactically annotated corpora of Estonian started at the end of 1990s with the training and test corpora for the Constraint Grammar shallow syntactic parser. By now the size of the Estonian Constraint Grammar Corpus is close to 300 000 running words. In 2004 the first attempts have been made to build deep syntactically annotated corpora (treebanks) for Estonian.

The Estonian treebanks have been annotated using the VISL formalism, developed at the University of Southern Denmark, which combines the phrase structure grammar with syntactic functions. There exist VISL treebanks for more than 20 languages already and they are used mainly for two purposes – for research and for education. Online visualization and query tools as well as edutainment software have been created within the VISL project to facilitate the both usages of treebanks.

Two small treebanks of Estonian were created as the joint work of University of Tartu and University of Southern Denmark. The research-oriented treebank Arborest has been semi-automatically derived from a section of the Estonian Constraint Grammar corpus and the Estonian teaching treebank was annotated manually. Currently, the Estonian teaching treebank consists of 100 sentences. The Estonian treebank Arborest contains 2500 sentences, 149 from which have been manually revised.

Keywords: annotated corpora, treebanks, syntax, Constraint Grammar, e-learning

1. Introduction

Data-driven methods are gaining more and more popularity and success in natural language processing. First, it is cheaper to let the computers discover the language rules instead of hand-crafting them, and, on the other hand, only the language software that contains probabilistic components, trained on the real data, could be able to cover the wide spectrum of texts produced by different language users and having different communicational goals.

For machine learning of syntactic rules of a particular language we need syntactically annotated corpora, also called *treebanks*. The term treebank comes from the graphical representation of a parsed sentence as a tree (cf. Figure 1). However, not only corpora containing tree-shaped representations are considered treebanks, but corpora with all kinds of structural analysis beyond the part-of-speech level, including semantic and discourse analysis (Nivre et al 2005). Treebanks can also be used for training and evaluation of morphological and syntactical analyzers and human language technology applications, for e-learning and for linguistic surveys.

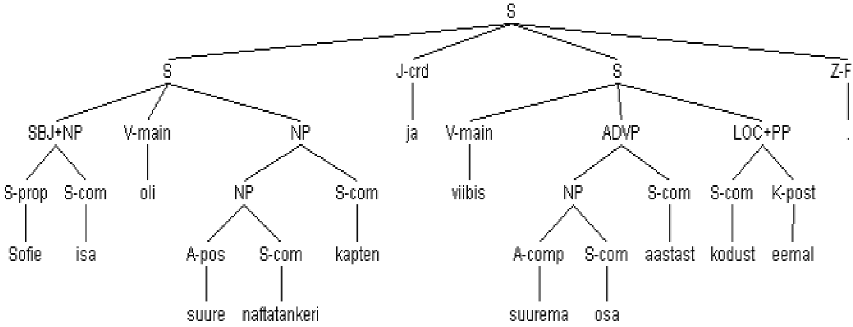


Figure 1. A sentence tree from the Estonian part of the Sophie Parallel Treebank

Creating a treebank is a time- and labour-consuming task. Therefore, it is important to look for the opportunities of re-using the existing methods and software for the treebank creation at its starting point. Ideally, the design of a treebank should be motivated by its intended usage, whether linguistic research or language technology development. However, in practice, there are a number of other factors that influence the design, such as the availability of data and analysis tools (Nivre et al 2005).

In 2003, before we started to create Estonian treebanks, we had a Constraint Grammar shallow syntactic parser for Estonian (Müürisep et al 2003) and some CG-annotated training and test material for the parser. In April 2003 the research group of computational syntax at University of Tartu joined the Nordic Treebank Network[1], a research network which aims at promoting treebank-related research in Nordic countries. The Estonian treebanks described in this paper have come into being thanks to the research cooperation between University of Tartu and other research institutions involved in this network. The most significant results have been gained thanks to the reuse of the method of semi-automatic creation of VISL treebanks developed at University of Southern Denmark (Bick 2003).

In this paper we will give an overview of treebank-related research activities for Estonian language. We will describe the creation process and annotation schemes of Estonian treebanks, which have been slightly different depending on the representation formats and purposes of the treebank. We will also look at the usage areas of syntactically annotated corpora of Estonian – research and e-learning.

2. Constraint Grammar Corpus of Estonian

The creation of syntactically annotated corpora of Estonian started at the end of 1990s. At that time we were developing a Constraint Grammar shallow syntactic parser for Estonian and some training and testing material was needed for that purpose. Depending on the funding the test corpus creation progressed in variable speed. By the time being the Estonian Constraint Grammar Corpus consists of ca 210 000 words of fiction, 80 000 words of newspaper texts and 6 000 words of legal texts. The corpus is

[1] http://w3.msi.vxu.se/~nivre/research/nt.html

stored as text files[2], each word on a separate line and its morphological and syntactical tags (Table 1) on the next line.

To use the existing treebank creation and usage tools it is often needed to make some kind of conversions on the corpus representation format. We have converted our syntactically annotated corpora to standard formats (NEGRA export format and TIGER XML) to facilitate the usage of treebank tools developed at University of Stuttgart within the TIGER project (Brants et al 2002).

Table 1. Estonian Constraint Grammar tag set

Syntactic function tag	**Meaning**
@SUBJ	subject
@OBJ	object
@ADVL	adverbial
@±FMV	finite (non-finite) main verb
@±FCV	finite (non-finite) modal or auxiliary verb
@AN>, @<AN	adjective as attribute
@NN>, @<NN	noun as a modifier (of a noun); apposition
@AD>, @<AD	adverb as a modifier (of a noun)
@Q>, @<Q	complement of quantor (*five* *men*)
@P> , @<P	complement of adposition (*on* *the table*)
@VN> , @<VN	participle as a modifier (of a noun)
@INF_N>, @<INF_N	infinitive as a modifier (of a noun)
@PN>, @<PN	adpositional phrase as a modifier (of a noun)

K. Kaljurand has written a Perl program for converting the CG corpus to NEGRA export format[3]. Now we can use the treebank creation tool Annotate and treebank search tool TIGERSearch to work on Estonian CG corpus. TIGERSearch is a powerful treebank search tool that imports grammatical structures from various formats, makes them searchable and displays them graphically. The graphical output of TIGERSearch is illustrated on Figure 2. Noticably, the CG annotated trees are very flat – phrase structure and the hierarchy of subclauses are not expressed.

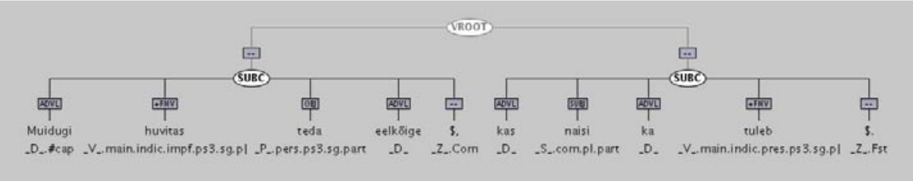


Figure 2. A sentence from the Estonian CG corpus in TIGERSearch graphical display format

There have been carried out some linguistically interesting experiments with Estonian CG corpus. Dependency relations were derived from the CG annotation, focusing on subject, object and adjective modification relations to build a database of

[2] http://lepo.it.da.ut.ee/~heli_u/SA/

[3] http://psych.ut.ee/~kaarel/Programs/Treebank/EstCG2Negra/

syntactically similarly behaving nouns. This database together with Estonian WordNet was used for word sense disambiguation (Kaljurand 2004).

3. Treebanks of Estonian

As the Constraint Grammar trees are too flat, we needed another formalism to represent the deeper structure of the sentence. The Estonian treebanks have been annotated using the VISL formalism, developed at the University of Southern Denmark, which combines the word-based shallow dependency tags with constituent trees (Bick 2003).

Each node in the VISL tree has two labels – a form label (e.g. fcl = finite clause, NP = noun phrase, VP = verb phrase, n = noun, adj = adjective etc.) and a function label (e.g. clause level syntactic functions like S = subject, O = object, P = predicate etc. and phrase-internal functions like H = head and D = dependent).

There exist VISL treebanks for more than 20 languages already and they are used mainly for two purposes – for research and for education. For both purposes small experimental treebanks of Estonian have been created.

4. Arborest – from shallow to deep syntax

The treebank Arborest was created semi-automatically. We have re-used the method of deriving phrase structure trees from the word-based Constraint Grammar annotation described in (Bick 2003). The Estonian treebank Arborest contains 2500 sentences, from which 149 have been manually revised. The evaluation of the automatically generated trees showed that 40 % of the trees were correct, i.e. had both correct branching structure and (almost) correct form and function labels.

Generalizing the study results, the correctly parsed sentence types were the following:

(1) simple sentences with the structures subject-predicate-object in any order plus optional adverbial(s) and predicative complement (C);
(2) composite sentences (subclauses bound with ja, ning, või, ehk or comma);
(3) complex sentences with subordinated clauses in the function of adverbial or object

Major sources of errors were adverbial attachment, non-finite subclauses, complex noun phrases and complex sentences with more than one subordinated clause.

More detailed description of creation and examination of Arborest is given in (Bick et al 2004). Estonian treebank Arborest together with a tgrep2-based search interface *tgrep-eye* is available at the webpage http://corp.hum.sdu.dk/arborest.html.

5. Online e-learning of Estonian syntax

VISL online edutainment games have been worked out at the University of Southern Denmark within the VISL project[4]. The games are based on the syntactically annotated corpora and are meant to learn the grammar of a particular language.

Using the VISL terminology some of the games are called "function games" (to learn syntactic functions) and others are "form games" (to learn word classes). The task to determine word classes (noun, verb, adjective etc.) has been implemented in different attractive ways in the games "Shooting gallery", „Labyrinth“„Wordfall“ a.o. In some other games, e.g. "Space rescue" the student has to determine the syntactic functions of words in a sentence.

[4] http://visl.sdu.dk

The teaching treebank of Estonian was annotated manually by H. Uibo, H. Nigol, K. Kerner and K. Nurmoja. It is a VISL treebank, as is Arborest, but we have decided to use a little different and in some aspects richer category set.

1. We have subcategorized the A (adverbial) tag:

- *fA* for clause level adjuncts
- *Ao* for NPs which are valency-bound to verb but are not objects as they are not in nominative, genitive or partitive case

2. As there exist both pre- and postpositions in Estonian, we invented a new tag *adp* for adpositional phrases (instead of *prp* used for tagging adp-s in the treebank Arborest).

3. Phrasal verb constructions are analyzed as a whole, using the tag *Vpart* as the function tag for the adverbial component.

Currently the Estonian teaching treebank consists of 100 sentences which are classified into the following ten complexity classes (10 sentences in each class):

1..10	S V and S V O (simple NPs, i.e. consisting from head only)
11..20	S V Adv and Adv V S (simple NPs)
21..30	like 1..20, but more complex NPs
31..40	S V O + adverbs allowed everywhere
41..50	like 1..40, but more complex VPs and predicatives (complements) allowed
51..60	simple sentences, having PPs in the adverbial function
61..70	composite sentences (subclauses on the same level in the tree)
71..80	complex sentences, subordinated clause in the function of object or adverbial
81..90	sentences containing coordinated NPs
91..100	sentences with non-finite subclauses

The teaching treebank and the VISL games for Estonian are available on the webpage http://beta.visl.sdu.dk/visl/et. A pilot project is carried on in some secondary schools to try out the e-learning of Estonian syntax in practice. We are going to revise the teaching treebank based on the feedback that we will get from the teachers and students. We are already aware of some difficulties of using the VISL games:

- User interface of the games is in English.
- Foreign words are used for word classes (in content, the material is suitable for 5.-7. year primary school students, but terminology is not known).

6. Conclusions and perspectives

By converting our corpora to the appropriate format we can reuse the tools developed for other languages. An important lesson learned in this process was that the use of standard encoding and annotation formats is crucial design decision from the very beginning.

In a longer perspective, we intend to extend the CG corpus to 500 000 running words, to correct and improve CG-to-PSG rules in the transformation grammar, and maybe refine the CG tagset (especially with regard to adverbial subclasses). A wider coverage of language variety is also desirable, and we would like to create spoken language treebanks for Estonian. How to represent dialogue act information etc is still open, but will hopefully be resolved in cooperation with the Nordic Treebank Network. On the teaching side, we intend to test the VISL games for Estonian in practice at some primary schools to get the feedback from students and teachers, and prepare the grounds for a more formal evaluation. Finally, our corpora are in continued need of manual revision, and a special focus area will be complexity classes of sentences, as well as adverbial subcategorization in both the Arborest and VISL teaching Treebanks. A special corpus initiative is the Estonian part of the Sophie parallel Treebank, which is being created using the methodology described for Arborest.

References

Bick, Eckhard 2003. A CG & PSG hybrid approach to automatic corpus annotation, In: Simow, K.; Osenova, P. (eds.) *Proceedings of SProLaC2003* (at Corpus Linguistics 2003) Lancaster. 1–12.

Bick, Eckhard; Uibo, Heli; Müürisep, Kaili 2004. Arborest – a VISL-style treebank derived from an Estonian Constraint Grammar corpus. In: Kübler, S.; Nivre, J.; Hinrichs, E.; Wunsch, H. (eds.) *Proceedings of the Third Workshop on Treebanks and Linguistic Theories (TLT 2004)*. Tübingen. 1–14.

Brants, Sabine; Dipper, Stefanie; Hansen, Silvia; Lezius, Wolfgang; Smith, George 2002. The TIGER treebank. In: *Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT 2002)*. Sozopol, Bulgaria. 24–42.

Kaljurand, Kaarel 2004. Word Sense Disambiguation of Estonian with syntactic dependency relations and WordNet. In: *Proceedings of ESSLLI-2004*, Nancy, France. 128–137.

Müürisep, Kaili; Puolakainen, Tiina; Muischnek, Kadri; Koit, Mare; Roosmaa, Tiit; Uibo, Heli 2003. A New Language for Constraint Grammar: Estonian. In: *Proceedings of International Conference Recent Advances in Natural Language Processing (RANLP 2003)*. Borovets, Bulgaria. 304–310.

Nivre, Joakim; de Smedt, Koenraad; Volk, Martin 2005. Treebanking in Northern Europe: A White Paper. In: *Nordisk Sprogteknologi. Nordic Language Technology. Årbog for Nordisk Sprogteknologisk Forskningsprogram 2000–2004*. Copenhagen: Museum Tusculanums Forlag. (forthcoming)

HELI UIBO is lecturer of language technology at the Institute of Computer Science, University of Tartu and member of the informal research group of Computational Linguistics, University of Tartu. She received her M.Sc. (computer science) in 1999 at the University of Tartu. Her research interests include Natural Language Processing, especially morphological and syntactical analysis of Estonian and creation and usage of syntactically annotated corpora (treebanks). She is the site coordinator in the Nordic Treebank Network (2003-2005), a research network funded by NorFA. E-mail: heli.uibo@ut.ee.

ECKHARD BICK works as research lector at the Institute of Language and Communication, University of Southern Denmark, where he is project leader of the VISL project. Eckhard Bick has degrees in Medicine (University of Bonn, 1984), Nordic Languages and Portuguese (cand.mag., Århus University, 1993). He defended a dr.phil.-thesis in Linguistics in 2000, also at Århus University, on Constraint Grammar based automatic analysis of Portuguese, and has since written Constraint Grammars for a number of languages. Current interests include corpus annotation, treebanks, grammar based spellchecking, computer assisted language learning, named entity recognition and question answering. E-mail: eckhard.bick@mail.dk.

DEVELOPING A FINNISH CONCEPT-TO-SPEECH SYSTEM

Martti Vainio, Antti Suni, and Paula Sirjola
Department of General Linguistics, University of Helsinki, Finland

Abstract

This paper presents the current phase of the development of a concept-to-speech system for Finnish. The system aims for a capability to produce prosodic contrasts pertaining to the information structure of the message and accepts as its input data structures, which are produced by a Natural Language Generator component and include such information as topic and focus as well as part of speech of each word. The system generates a paragraph or speaking turn sized utterance sequences which can be related to both the discourse and information structure of the given situation. We aim for a very generic system which is not restricted to a given application such as, for instance, a spoken route description. The prosody control and signal generation of the synthesizer are based on a Hidden Markov Model based system (HTS). The paper describes the overall structure of the system. We will also present the mark-up required for the training material as well as preliminary results from applying the synthesis method to a database originally developed for text-to-speech research.

Keywords: concept-to-speech, speech synthesis, Finnish prosody

1. Introduction

The development of concept-to-speech (CTS) systems are usually justified by increased prosody control stemming from the fact that the linguistic structures on which the synthesized speech is based are generated from scratch rather than inferred from text. The generation process is less prone to error and usually the system responsible for the generation possesses *more* information than can be analyzed from text. Namely, the systems know the meaning of the generated linguistic structures as well as their informational status; which information is new and which is given. This gives them increased control as to which words should be emphasized more than others. In other words, concept-to-speech systems should be able to produce proper prosody for a given situation and do so without the fear of miss-leading the receiver.

Far less frequent are justifications based on basic research regarding speech prosody. Speech synthesis has always been a tool for doing basic phonetic research and text-to-speech (TTS), in particular, has lead to many insights into prosody. In this vein, concept-to-speech can be seen as an even more valuable research tool – after all, there are no unnatural constraints forced on the systems by the lack of proper tools for analyzing text. Even the lack of a full-fledged natural language generation (NLG) component can be bypassed by template-based language generation. The concept of a CTS system as a research tool is of consequence especially with regard to languages whose prosody – especially intonation – is not well known.

Typically CTS systems extend the capabilities of TTS systems by adding discourse and information structure level as well as semantic information to the synthesizer input. As mentioned above, apart from being more abstract and on a higher level with regard to linguistic analysis, they can be added to the input in a confident manner as the linguistic structures are generated. What information exactly is needed varies depending on the intended use of the system as well as the methodology chosen for the actual prosody control of the system. Usually the pitch accent prediction has been separated from the phonetic implementation that produces the eventual pitch curves for the utterances. This is also the case in our CTS system where the pitch accent prediction is done in the prosodic front-end of the synthesizer and the f_0 contours are generated in the phonetic module.

Pan and McKeown (1998) have shown that pitch accent prediction can be done in a reliable manner with only four semantic and syntactic features. On the other hand, Hitzeman et al. (1999) have shown that the inclusion of NP related and rhetorical information reduced the error in pitch accent prediction by 15.5%. What previous research on CTS has shown is, that it is not as straight-forward as is may seem to choose the information for controlling prosody. It is, however, important that the features used are as abstract as possible. This will ensure that the system is of maximal use regardless of the application domain.

In what follows we will describe the first phases of the development of a CTS-system for Finnish. First the overall design of the system is described followed by a presentation of results from evaluating the system with data designed for TTS-research and conclude with discussion on future work.

2. Synthesizer Design

The synthesizer consists of three main modules: the *linguistic front-end* responsible for either generating the linguistic structures from input data or analyzing text into similar linguistic structures; the *prosodic front-end* responsible for generating the linguistic prosodic structure; and the *phonetic module* responsible for the actual speech signal generation (including f_0 values and segmental durations).

The system accepts as its input data which the NLG component will render into an XML based format which includes all necessary information for the linguistic components of the synthesizer to work on. The possibility to achieve a high quality syntactic analysis by *Functional Dependency Grammar* (FDG) syntactic tagger (Tapanainen and Järvinen 1997) enables the system to render high quality speech from free text input, as well. The training data is also automatically tagged with the FDG component (in addition to the manually added prosodic tags).

2.1. Prosodic Front-end

Using the Python programming language, we have implemented what we call a prosodic front-end with XML as an internal representation of the utterance structure. The purpose of this module is to bring together the information from both the NLG and FDG components and render it to the format which is suitable for the phonetic module. In its current state, the front-end consists of conversion from FDG-parser output, information content (IC) assignment, tracking of new - given status of nouns and rule-based syllabification, letter-to-sound conversion, as well as stress and accent assignment. Currently the accent prediction is done with a set of hand-configured rules which use part-of-speech, IC and syntactic constituent structure for making decisions about accentuation on four distinct levels. The final shapes of the intonation contours are predicted statistically in the phonetic module.

The status of XML as a standard markup language of structured documents makes it easy to convert and consequently synthesize linguistically enriched documents from various sources. The tree structure of the utterance is also well suited for the XML-model.

The prosodic front-end is used for feature extraction from arbitrarily chosen unit attributes and positions in the utterance tree to produce contextual label sequences in a format suitable for the phonetic module at both training and synthesis phases. The system is furthermore used for producing optimized *questions* for the decision trees used by the phonetic module before the training phase. The questions are compiled using descriptive statistics of the training data. The XML tree structure produced by the front-end can also be used in quantitative evaluation of the system's performance. That is, global RMSE and correlation estimates can be further broken down by assigning these measurements to units of lower levels in the utterance tree. For instance prediction errors can be calculated for syllables, morae, and words and these can be further chosen by any desired linguistic or structural criteria.

2.2. Phonetic Module: HMM-Based Speech Synthesis

HMMs have been a standard model for automatic speech recognition (ASR) for several decades, but only relatively recently has it been shown that the models can also be applied to high quality synthetic speech production. The Nagoya Institute of Technology have released a set of modified HTK tools (Young 1993) and a signal generation engine, which we have adopted for our CTS framework (HMM-Based Speech Synthesis System, HTS, `http://hts.ics.nitech.ac.jp/`).

Conceptually, the training of HMM-models in the HTS is similar to training a typical HMM-based speech recognizer. Spectral and pitch parameters are first extracted from acoustic training data. Monophone HMMs, composed of cepstral and log-f0 streams with state-duration densities, are trained using time-aligned phone labels. Monophones are then used to initialize the context-dependent HMMs, defined by contextual labels generated using the front-end. Context-dependent models are then further refined. As a large number of contextual features are used, the training data lacks most of the feature combinations and the HMMs can not be trained robustly. Therefore, decision tree clustering is performed constrained by phonologically or otherwise motivated questions provided to the system by the developer. As spectrum, f_0 and duration have their own influential features, these parameters are clustered independently (Yoshimura et al. 1999).

In the synthesis phase, utterances in the form of contextual label sequences are provided as an input to HTS. An utterance HMM is created by concatenating the context-dependent HMMs, which are selected using the decision trees. Speech parameters are then generated from the HMM using the algorithm presented in Tokuda et al. (2000), applying dynamic features to constrain transitions in the phone boundaries. Finally, speech is generated from the parameters using a *mel-log spectral approximation* filter.

3. Preliminary Tests and Results

The training of the system was done with speech material originally designed for research on text-to-speech synthesis containing isolated sentences (Vainio 2001). This type of material usually lacks prosodic features which are related to information and discourse structure. There are, however, a number of utterance internal prosodic phenomena that we wanted to be able to model before moving into higher level data which would better represent what is required of a CTS-system.

We trained three distinct systems with a varying amount of input information for prosody control. The baseline system (POSITION) was trained with phone identities and quantities in a four phone window and 30 positional and quantitative features ranging from phone position in syllable to number of phrases and phrase position in the utterance. The system trained with accentuation was evaluated by predicted (ACCENT) and actual accentuation (Pred. ACCENT, obtained from manually tagged data). The most simple system (PHONE) was trained with using only phone identities and quantity information (four-phone window). The results are summarized in Table 1.

3.1. Training and Testing Data

The training corpus consists of 692 declarative Finnish sentences containing 6455 words spoken by a male speaker. 600 utterances were used for training and the rest for testing. The corpus was manually segmented and phonetically transcribed. We enriched the transcription by manually marking the accentuation on four distinct levels (from unaccented to strongly emphasized). Phrase boundaries were also introduced when they were not predictable from punctuation.

Acoustic training material was prepared for training by extracting f_0 curves with ESPS-Waves pitch-tracking algorithm adjusted to speaker's range; no post-correction was used. Mel-cepstral analysis was performed with the *Speech Signal Processing Toolkit* (SPTK, `http://kt-lab.ics.nitech.ac.jp/~tokuda/SPTK/`). We used frame length of 5 ms, and 25 cepstral coefficients were extracted for each frame, together with their dynamic features. 5-state left-to-right HMMs were used in training.

3.2. Results

Quantitative evaluation was conducted for the four test configurations by comparing duration and pitch of synthesized test utterances with the original utterances. For pitch evaluation, synthesis was performed with reference phone-level time alignment, and F0-values of both original and synthesized data were extracted using 5ms frames. Those frames where f_0 value were found for both test and reference were used in calculations. Utterance-wise normalization was performed by subtracting the mean values of pitch and duration values.

Standard measurements of RMSE and Pearson's correlation were used to estimate the performance of the systems in relation to each other. Values for the three first morae of each word were used in calculations, as differences are most marked around stressed syllables. Standard deviation for pitch and duration of the original test utterances were 2.89 semitones and 0.033 seconds, respectively.

Figure 1 shows word-level segmentations and f_0 curves for the original utterance as well as one predicted by a system which has been trained with manually marked accentuation values. It can be seen that the system is responding correctly to the input accentuation information (see especially the attenuated accents marked with “1” in the transcription. The higher beginning caused by a new topic is also clearly seen as are the typical “down-stepped” accents (marked with “2”). There are still some relatively low level timing discrepancies which are probably due to concentrating on syllables rather than morae as the domain of accentuation (Suomi et al. 2003).

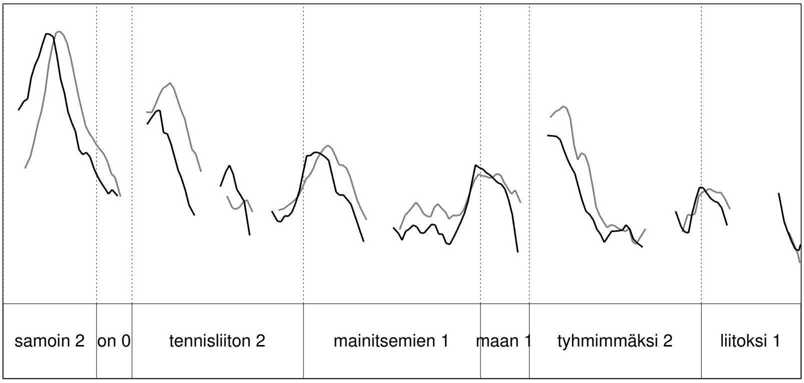


Figure 1: An original f_0 curve (grey line) and a synthesized curve (black line) of the beginning of the utterance “Samoin on tennisliiton mainitseminen maan tyhmimmäksi liitoksi kohtuuton ja väärä” (As is mentioning the tennis association as the country’s most stupid both unfair and wrong.). The manually transcribed accentuation levels are marked after each word.

Table 1: Results for four distinct systems trained using different input configurations. .

	f0, semitones		duration, seconds	
	RMSE	r	RMSE	r
PHONE	2.66	0.409	0.0246	0.676
POSITION	1.93	0.760	0.0219	0.756
Pred. ACCENT	1.87	0.780	0.0217	0.760
ACCENT	1.82	0.794	0.0219	0.757

4. Conclusion and Future Work

We have shown that the HTS system is well capable of producing prosodic phenomena from low level prosodic input such as accentuation and low level structural information such as phones, syllables, and morae as well as positional and quantitative information such as number of phrases, word and syllables in the utterance. What remains to be done is to identify the linguistic, structural, and prosodic features that are related to contrastive focus and other discourse related phenomena and integrate them to the synthesizer in such a manner that they can be controlled with high level abstract tags. While doing this we have to simultaneously observe how those phenomena are manifested phonetically and tag our corpora accordingly. In this way we have to iterate until we have hopefully separated the phonological units from their phonetic realizations. It is in this way that the CTS system becomes an indispensable research tool for doing basic research.

5. Acknowledgments

This research has been financed by the National Technology Agency of Finland (Tekes) with the Fenix research program, subprogram PUMS 2 (Puheteknologian uudet menetelmät).

References

Hitzeman, Janet; Black, Alan W.; Taylor, Paul; Mellish, Chris; Oberlander, Jon 1999. An annotation scheme for concept-to-speech synthesis. In: *Proceedings of the European Workshop on Natural Language Generation*, Toulouse, France. 59–66

Pan, Shimei; McKeown, Kathleen 1998. Learning intonation rules for concept to speech generation. In: *COLING-ACL*. 1003–1009

Suomi, Kari; Toivanen, Juhani; Ylitalo, Riikka 2003. Durational and tonal correlates of accent in Finnish. In: *Journal of Phonetics* **31**, 113–138

Tapanainen, Pasi; Järvinen, Timo 1997. A non-projective dependency parser. In: *Proceedings of the 5th Conference on Applied Natural Language Processing*. 64–71

Tokuda, Keiichi; Yoshimura, Takayoshi; Masuko, Takashi; Kobayashi, Takao; Kitamura, Tadashi 2000. Speech parameter generation algorithms for HMM-based speech synthesis. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey,*: Vol. 3. 1315–1318

Vainio, Martti 2001. Artificial Neural Network Based Prosody Models for Finnish Text-to-Speech Synthesis. No. 43 in Publications of the Department of Phonetics, University of Helsinki, Yliopistopaino

Yoshimura, Takayoshi; Tokuda, Keiichi; Masuko, Takashi; Kobayashi, Takao; Kitamura, Tadashi 1999. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In: *Proceedings of European Conference on Speech Communication and Technology, Budapest, Hungary*: Vol. 5. 2347–2350

Young, S. 1993. The HTK Hidden Markov Model Toolkit: Design and Philosophy. Technical report

MARTTI VAINIO is currently employed as a university lecturer at the Department of General Linguistics as well as the Department of Speech Sciences of the University of Helsinki. He received his M.A. (Phonetics) the University of Helsinki, Deparment of Phonetics, dealing with Finnish text-to-speech synthesis. His doctoral dissertation (also at the Department of Phonetics) dealt with prosody control with artificial neural networks for Finnish text-to-speech synthesis. He has taught speech technology, phonetics and speech acoustics related courses at the University of Helsinki and at the Helsinki University of Technology. E-mail: martti.vainio@helsinki.fi.
ANTTI SUNI is a computer science major at the University of Helsinki and is currently preparing a master's thesis within the project. He has many years' experience in developing a commercial ASR system for Finnish.
PAULA SIRJOLA is a language technology major at the University of Helsinki and is currently preparing her master's thesis in the project.

III Posters

A GRAPHICAL QUERY FORMATION COMPILER FOR SPEECH DATABASE ACCESS

Toomas Altosaar[1]**, Mietta Lennes**[2]
[1]Laboratory of Acoustics and Audio Signal Processing,
Helsinki University of Technology (Espoo, Finland)
[2]Department of Speech Sciences, University of Helsinki (Helsinki, Finland)

Abstract

This paper presents a system where a desired query pattern in a speech corpus is formed graphically. The ensuing structure is then automatically compiled into executable code that can be readily applied to object models of speech. Speech database queries can thus be defined in a high-level manner both intuitively and interactively. Queries are graphically specified by creating matching template units that are instances of classes, e.g., segments, phones, phonemes, syllables, and words. Also, the features and properties of these instances are defined, e.g., voicing, part-of-speech tags, etc. Types, features, and properties of each unit can be made as specific or general as required, enabling different degrees of query selectivity. The paper presents the system using actual queries on part of a Finnish speech database.

Keywords: speech databases, object-oriented, graphical query, automatic code-generation, compiler.

1. Introduction

Speech corpora form a basis for research on spoken language since theoretical hypotheses can be tested comprehensively on large volumes of data. Once a speech corpus has been richly annotated, it can provide large quantities of information about complex phenomena that might not be discovered with any other method. Currently, many spoken language corpora are already available. However, before corpora can be used, an infrastructure is necessary that facilitates access to the data. A speech database – corpora integrated with an access mechanism – fulfills this requirement (Hendriks 1990). Queries can then be applied to reveal the complex relationships between the different units that make up speech.

Speech database access is typically performed on relational database management systems (RDBMS). Here the rich structure of speech first needs to be flattened due to the inherent representational constraints of the RDBMS paradigm. In an object-oriented database (OODB), a more natural model of speech can be formed thus allowing complex queries to be designed and applied reliably (Altosaar et al. 1999).

Typical queries in an OODB system may take the form of functions that search the local environment of a speech utterance, moving repeatedly from unit to unit and testing for matching contextual locations. These query functions may return matches according to some binary or continuous valued function, e.g., all [A] phones in some context are returned as a set of matches, or, all units that the query was applied to are returned but sorted according to some closeness-of-match criterion.

However, specifying these query functions textually, such as with computational methods in a programming language, e.g., C, Lisp, etc., usually requires that the user be fluent in the query syntax or implementation language of the speech database. This has the negative effect of restricting the number of potential users being able to utilize the system. An intuitive and easy to use interface would have much value since design and manipulation of speech database queries could be performed by non-programmers. By lowering the required skill threshold, meaningful research could be performed, e.g., by students and people working outside the core speech technology area. Other benefits would include the possibility to test and verify the related annotations of a corpus since queries could be formed interactively and tested immediately on parts or entire volumes of speech data. Syntax and validity checking of queries by the system is also important since faulty queries could be detected before their expensive application to large databases would commence.

Queries may be realized in different ways as well. If different annotation layers are supplied in orthographic/phonetic form for words, syllables, phones, etc., then searches can be performed for the annotations using string-based regular expressions. A typical task could be to find all occurrences in a corpus for a consecutive sequence of phones, e.g., [k][a][C+][a] where [C+] indicates one or more occurrences of any consonant phone. However, for corpora where object models have been built, string-based searches may be an insufficient search paradigm since not all data can be mapped to strings and subsequently tested. For example, a similar task in an enriched object-model would be to find the same pattern with the additional constraints that the F0 of the first [a] is [20-25] % higher than the F0 of the second [a], and that a [5-10] % declination of pitch is occurring over a three-word window. This type of complex query might prove to be substantially more difficult to define without the availability of an object model. Subsequently, in object models of speech it is useful to envision the process of querying a corpus as an exercise in structural pattern matching.

Such structural queries may take the form of functions that search the local environment of a speech utterance, moving repeatedly from unit to unit and testing for matching contextual locations. These functions are usually defined by an experienced user who must be fluent in the system's language of implementation, e.g., C++, Lisp, etc. Furthermore, defining these types of complex queries can be a highly error prone and difficult activity since the correctness of searches may be difficult to prove leading to expensive reapplications of queries over large corpora.

This paper presents the design of an interactive system for forming and manipulating queries that can be applied to speech databases. Through visualization, the reliability of query specification is expected to be enhanced in the sense that fewer human errors are made when compared to writing equivalent query functions by hand in a textual format. The interface under development is integrated with the QuickSig speech OODB system (Altosaar 2001). Speech corpora that have been represented under QuickSig include, e.g., TIMIT, ANDOSL, Kiel, Estonian and Finnish.

2. Object Model for Representing Speech

Representing data with objects allows abstraction layers to be created that hide implementation details. Focus can move from low-level issues to higher-order concepts that are more directly related to the problem at hand.

Speech offers itself to be modeled readily with objects based on classes, e.g., classes can be defined for items such as a signal, word, phoneme, segment boundary, a talker, etc. Objects, which are instances of these classes and strive to model actual data, can be associated with other objects, e.g., a talker may have produced a set of signals, with each signal having one or more annotations, which in turn may include sentence, word, syllable, and phone (i.e., speech sound) objects. Objects can be made aware of their local surroundings using links enabling efficient automated inferences to be performed during query stages. Groups of linked objects representing, e.g., the same utterance, are referred to as a *representation framework.*

2.1 Queries

Search spaces are typically sets of frameworks, e.g., an entire corpus, or several corpora. Matching contexts can be found in a representational framework by first defining a *query template* that resembles a fragment of a larger structure, as shown in figure 1. Database search takes place by measuring the closeness-of-match between a query template and local framework structure. For each comparison, a distance measure is calculated that indicates how well the template matched part of the framework. Typically, the distance measure returns a simple binary valued result: either the query template matched the structure at some location or it did not. The result of a query is a set of matching locations represented as a list of pointers to speech units in the frameworks. The results can be thus used in further speech processing tasks efficiently since they offer a low database impedance mismatch (Altosaar & Vainio 2000) with the application, i.e., the results of the query and the speech processing application share the same data structures.

Example 1: Combinations of phone segments

The goal is to locate all fricative-vowel phone pairs in one or more frameworks. This query can be envisioned as a simple dipole query template structure, consisting of the two phone units shown in figure 1.

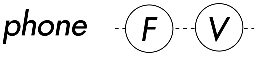


Figure 1: Example query template for identifying locations of fricative-vowel [F-V] phone pairs.

For an example search space, figure 2 shows a partial framework depicting the phonetic annotation for the Finnish sentence *isompi on suurempi*, where sentence, word, syllable, and phone speech units exist.

By visually inspecting the query template and the framework to which it is applied, one notices that two locations fit the match: the [s-o] and [s-u:] pairs. This is since [s] can be thought of inheriting from the more general fricative class, and both [o] and [u] from the vowel class. By specifying the desired quantity of phonemic units, e.g., short or long, this query template could be made even more selective if required.

2.2 Queries Expressed in Lisp

In order to automatically locate desired locations in an utterance of speech, small sections of programming code have been written to model these queries. This code is then applied to a framework's existing units, or a subset of these, and an index calculated indicating how well the query matched the framework at some location. The QuickSig speech OODB system relies heavily on class hierarchies and automatically builds frameworks prior to query application. The relationships between different

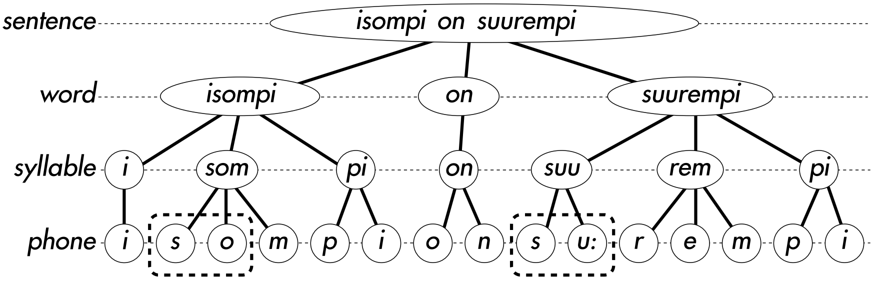


Figure 2: Part of a framework for an example utterance. Boxed areas show matches for the query template of figure 1. The Finno-Ugric phonetic alphabet is used on the phone level; other levels are marked with orthography.

annotated units and their properties can be defined according to the user's needs and can be expressed in some machine-readable formalism (e.g., RDF/XML) and then represented in the structure.

For example, in Lisp, the implementation language used in the QuickSig OODB system, the query of figure 1 can be written as:

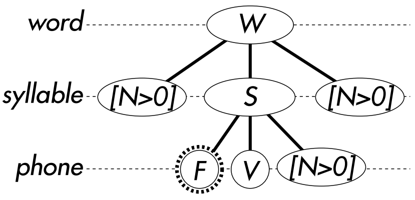
```
(lambda (x)
  (and (typep x 'fricative)
       (typep (next-unit x) 'vowel)))
```

Here *lambda* defines a function and the variable *x* represents an arbitrary unit in the framework where query *focus* is currently positioned, i.e., the unit being currently tested. The *and* macro requires that all of its arguments are not *nil* valued for it to return a non-*nil* value (*nil* signifies false in Lisp). The *typep* function tests whether its first argument inherits the class specified by the second argument (a symbol) in its class hierarchy. For example, the desired context requires the first part of the phone pair query to be a fricative, so the variable *x* is tested whether it has the class *fricative* in its class hierarchy. For the template to match part of the framework, the second phone must be a *vowel* and this is achieved by testing the *next-unit* of *x*. Predefined functions exist for units in the frameworks that can be applied generically and utilize the links in the frameworks to access local context. For example, the functions *prev-unit* and *next-unit* access adjacent units on the same level, the function *units* returns a list of all lower-level units, the function *unit-of* returns the parent unit(s), etc. These functions also accept additional arguments for inter-domain navigation, e.g., between the phonetic, phonemic, orthographic, acoustic, etc. domains.

However, as the complexity of the query increases so does the complexity of the textual query. This can be seen if we add the following two conditions to the query template of figure 1:

- the phone pair should exist at the beginning of a syllable (refer to this syllable by the variable *s*)
- *s* is neither a word-initial nor a word-final syllable.

Figure 3 shows the modified query template required to express these two additional constraints.



```
(lambda (x)
  (and
   (typep x 'fricative)
   (typep (next-unit x) 'vowel)
   (let* ((s (unit-of x))
          (w (unit-of s))
          (syllables (units w)))
     (and
      (eq x (first (units s)))
      (neq s (first syllables))
      (neq s (first (last syllables)))))))
```

Figure 3: Original query template with additional constraints. Units marked with bracket notation are wild-card units, e.g., at least one syllable unit must be present before and after S, and at least one phone must follow the [F-V] pair. Query focus is positioned on the fricative phone unit F and is indicated by a dotted circle.

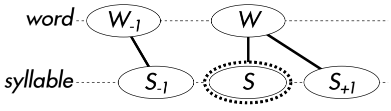
By referring back to figure 2 it can seen that the new query continues to accept the first match from the original query, but rejects the second match since both additional contextual constraints are not fulfilled by the [s-u:] pair. The program code that represents this new query template is shown on the right-hand side of figure 3.

The initial part of this query is similar to the one for figure 1. Since access is needed to all of the syllables in the word, the *let** construct sets up three temporary variables: *s* is a syllable in which the potential [F-V] pair resides, and *s* is then queried for its parent word *w*. The variable *syllables* is a list of all syllable units within *w*. With this information, an extra *and* form is now able to check for the additional constraints: the function *eq* is a predicate function that checks whether *x* (now known to be a fricative) is the same object as the first unit in *s*, and that *s* should not be the first syllable in the word, nor the last one. The *let** construct finally returns the result of the additional constraints (either a non-nil or nil value) and thus participates equally with the initial [F-V] constraint. When applied to the framework in question, only the [s-o] phone pair is found to match the new query this time and a list containing one element is returned: a pointer to the first [s] phone unit in the framework. The reason why the phone unit [s] is returned and not a word or a syllable unit is because the query focus was set up to focus on fricative objects solely.

Example 2: Searching for pitch patterns

The user might be interested in how the fundamental frequency (F0) contour reflects the perceived prominence or accent within utterances. Since F0 is considered to be an important factor in prominence perception, a database query could be performed, e.g., by matching a template where the middle of three consecutive syllables should have a higher F0 value than its immediate neighbors, see figure 4. In order to obtain comparable F0 values, the median of F0 samples within the vocalic part, i.e., the nucleus, of each syllable is calculated. Finnish does not exhibit lexical stress, but the word-initial syllable is likely to be perceptually the most prominent in accented positions. Therefore, the template focus is set to match word-initial positions only.

The matching units are post-processed by calculating the local second derivative of pitch that serves as an *accent index* and represents the psycho-acoustic magnitude of prominence of the syllable unit in focus. Therefore, a list of matches is returned composed of syllable objects along with the F0 values and accent indices. The values are mapped onto the semitone scale which closely corresponds to perceived pitch. The



```
(lambda (x)
  (and (typep x 'phonetic-syllable-unit)
       (zerop (unit-position (superstructure x) x))
       (let* ((f0_x (f0 x))
              (x-1 (prev-unit x))
              (f0_x-1 (f0 x-1))
              (x+1 (next-unit x))
              (f0_x+1 (f0 x+1)))
         (and f0_x f0_x-1 f0_x+1
              (> f0_x f0_x-1)
              (> f0_x f0_x+1)))))
```

Figure 4: Query template for syllable patterns where the central syllable has a higher pitch than its neighbors as well as the same predicate function expressed in Lisp. The function *zerop* returns a value of true if its argument is zero, unit-position returns an integer indicating the index of unit while *superstructure* is another way of expressing *unit-of*.

magnitude of accent can now be readily compared among different speakers and linguistic/phonetic contexts. Since the list contains objects that have direct links to their original locations in the annotated speech corpus, further queries can be applied to the syllables. Additionally, the original sound signals, possibly with some expanded temporal context, can be played back to the user. Moreover, in case the user wishes to run perception tests in order to check his or her newly formed hypotheses, the corresponding audio signals can be extracted from the database.

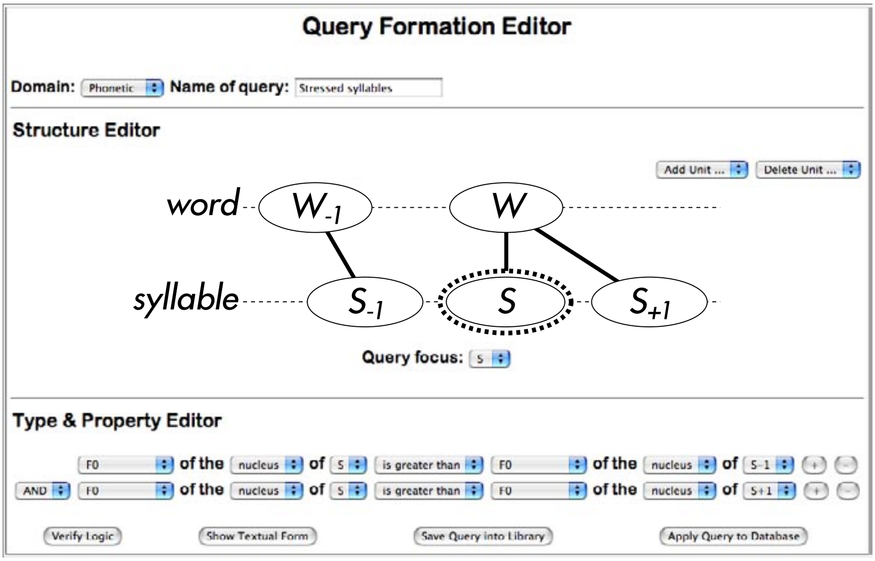
As seen by these examples, defining queries in the implementation language of the OODB allows for high degrees of expressive freedom. Complex queries, e.g., those including recursion, or structural elements that are defined to be of a wild card matching nature, can also be written in textually represented program code. Some criteria that we have found to be important, and existing in a query formalism, are the abilities to:

- access local context through the use of navigational functions, e.g., *prev-unit*, *next-unit*, *units*, *unit-of*, etc., that utilize explicit or computational links in frameworks,
- associate symbolic names with query templates so that libraries of queries can be published and reused by other queries and users,
- define local functions within a query so that recursive search methods can be applied.

This textual formalism is however error-prone to the programmer and cryptic to the non-programmer. Learning to use a specific textual query syntax or programming language presents a substantial usage barrier in any case. If query structures could be created visually, e.g., on a standard html-compliant browser, then this would improve robustness, address the learning-curve issue, and provide a portable implementation base as well.

3. Design of the User Interface

Retaining the expressive power of textual queries in a user interface requires several key components. These elements must enable and aid the user in forming the necessary constraints in several different query dimensions, e.g., structural, type, property, and symbolic references. The function and design of the components of a query formation interface are now presented.



Query Formation Editor

Domain: Phonetic Name of query: Stressed syllables

Structure Editor

Query focus: S

Type & Property Editor

F0 of the nucleus of S is greater than F0 of the nucleus of S-1

AND F0 of the nucleus of S is greater than F0 of the nucleus of S+1

Figure 5. Depiction of a possible user interface that is designed to fulfill design constraint specifications.

3.1 Structural Editor

This part of the query specification is used to generate only the query's structural shape, i.e., refer to only the shapes of figures 1, 3 and 4. The user begins by first selecting a domain and level from a menu and then creates a single unit. Parents, siblings, and children can then be subsequently added to this unit or other units to create the desired template structure. Query template units can be defined over many domains and linked to one another if necessary to form multi-dimensional queries (Altosaar 2001). For this reason it is advantageous that the graphical interface of the structural editor supports 3-D visualization. The top pane of figure 5 shows one possible user interface for a structural editor. A single mouse click either selects or de-selects a unit. In this example S is set as query focus.

3.2 Symbolic References

A symbolic name can be assigned to a selected unit and published so that other query structures can reference it. Named queries can be freely modified, combined, and saved to form new queries that can be used by other users. For example, if the user saved the [F-V] phone pair query of figure 1 into a library, this could be attached to unit *s* of figure 3 in one editing step.

3.3 Type and Property Editor

The internal behavior of template units is defined through a type and property interface. For example, a unit may be required to be a phone (i.e., a speech sound) such as [A], or a less specific unit such as a back-vowel, vowel, or any phone. A word unit could be selected on the basis of its morphological properties, in case these have been annotated.

Properties can also be temporal or acoustic, e.g., the duration of a segment, or the fundamental frequency (F0) calculated from a sound fragment that corresponds to a selected unit. Types and properties are assigned to a unit by constructing a logic tree of any breadth or depth. Multiple constraints may be assigned to a single query unit as well, e.g., a template unit may be specified to be a fricative *or* a tremulant. Other properties of units, such as part-of-speech tags, can be assigned to applicable units in a similar manner. Required query complexity can thus be captured, see the lower pane of figure 5. Operators can be any Boolean or analog function.

3.4 User Interface

Figure 5 shows one of many possible designs for an interface that enables users to generate queries in a high-level and intuitive manner. While queries are being formed they can be applied to a database immediately and the number of matches indicated to the user. Matches can be passed on to other speech processing tasks directly, e.g., speech synthesis or recognition algorithms, without the need for inefficient temporary file-system operations. Within the OODB QuickSig system the results can also be visualized as interactive graphs at intermediate query stages, see example in figure 6.

3.5 Compiler

As the user modifies and shapes the query in terms of structure, type, property, and symbolic references, the system compiles the graph from the focus unit and produces an object that is a single argument function. The function represents the conditions that have to be met for an element to “match” the search query. The function can be envisioned as a selectively tuned filter that rejects contexts where all of the conditions are not satisfied. Computationally searches are efficient since as soon as any one of the conditions are not met, the system will immediately abandon trying to match the current context and move on to the next unit to be tested.

Automatic compilation is accomplished by a graph walking algorithm that traverses the graph in the structure editor and collects the structural constraint of each unit with respect to one of its neighbors. Once all of the units have been visited, a traversal back to the focus unit is made in order to collect the type and property specifications. The final form represents the structural requirements of the query.

Symbolic references are resolved in a similar manner. Whenever a symbolic reference is found for a node it is expressed in terms of the focus node. Thus expressions containing symbolic references are resolved before supplying them to Lisp’s compiler, e.g.,

(> (F0 S) (F0 S_{+1}))   =>   (> (F0 x) (F0 (next-unit x))

In figure 6 only syllables with a local maxima in terms of F0 are shown: syllable duration (y-axis: from 50 ms to 0.6 seconds) vs. pitch accent index (x-axis: semitones). Material analyzed was 117 sentences read aloud by two male speakers: speaker 1's stressed syllables are in dark, speaker 2 in light. From this plot of syllable objects it can be seen that speaker 1 has produced his "accented" syllables with smaller pitch changes than speaker 2, since no dark markers can be found above the accent value of ca. 7.5 semitones. This finding is consistent with the perceptual impression of the speakers' voices. As for the duration of the accented syllables, the two speakers apparently behave in a similar fashion, i.e., they have produced the syllables at approximately the same speaking rate. Plotted markers are also mouse sensitive: a click on a marker causes the temporal region around the matching syllable to be played as audio.

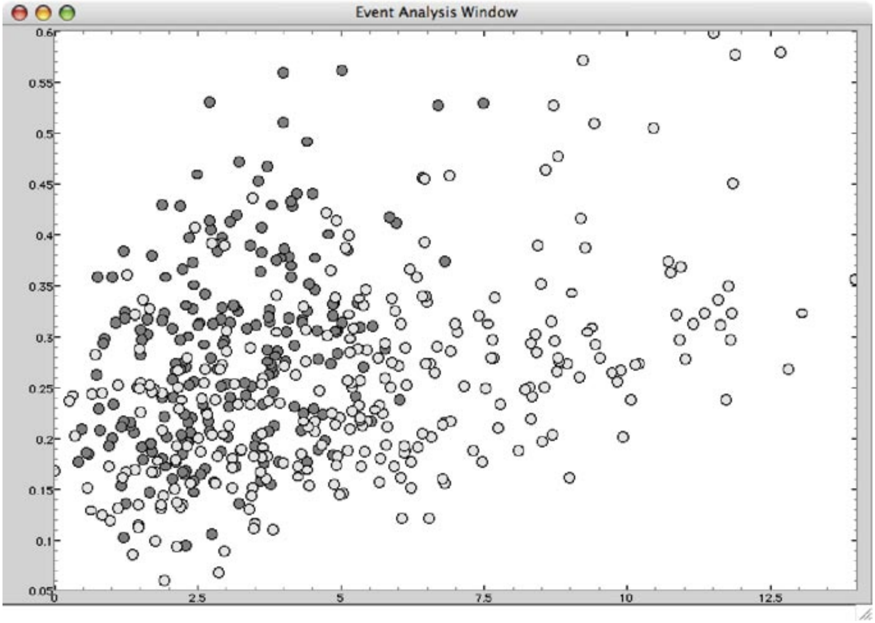


Figure 6. Query matches can be visualized to check the effects of different constraints.

4. Conclusions

This paper presented the design of an interactive system for forming and manipulating queries that can be applied to speech databases. Through visualization, the reliability of query specification is enhanced since textual queries are generated by a graphical compiler. Fewer human errors are thus introduced when compared to writing equivalent query functions manually in a textual form. Furthermore, users need not be fluent in the implementation language of the underlying system thereby opening up corpora usage to a much larger audience of phoneticians and spoken language researchers.

Acknowledgments

This research has been funded by both the TEKES (The National Technology Agency of Finland) project entitled “New Methods and Applications of Speech Technology” (40398/04), as well as by the eVikings-II project (FP6 IST-2001-37592).

References

Altosaar, T., Millar, B. & Vainio, M. (1999). Relational vs. Object-Oriented Models for Representing Speech: A Comparison Using ANDOSL Data. In Proc. of Eurospeech-99 pp. 915-918. Budapest, Hungary.

Altosaar, T. & Vainio, M. (2000). Reduced Impedance Mismatch in Speech Database Access. In Proc. of the ICSLP-2000. vol. 1. pp. 778-781. Beijing, China.

Altosaar, T., Object-based Modelling for Representing and Processing Speech Corpora. Report no. 63 / Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing. Espoo, Finland, 2001.

Hendriks, J. (1990). A Formalism for Speech Database Access. Speech Communication, 9, pp. 381-388. Elsevier Science Publishers B.V., North-Holland.

TOOMAS ALTOSAAR is a researcher at the Laboratory of Acoustics and Audio Signal Processing at the Helsinki University of Technology (HUT). He received his B. Eng. degree in Electrical Engineering from McGill University, Montréal, Canada, a Licentiate degree in Electrical Engineering, a EuroMBA diploma, and the Doctor of Science (Technology) degree, the latter three from HUT. He has worked in the field of speech processing for more than 20 years being involved in a wide range of speech technology related fields including speech recognition and synthesis, spoken language databases, and speaker recognition/verification systems. He has authored over 50 international publications in the area of signal analysis and processing algorithms, neural networks, and object-oriented databases. E-mail: Toomas.Altosaar@hut.fi

MIETTA LENNES is a phonetics researcher at the Department of Speech Sciences of the University of Helsinki, Finland where she received her M.A. in 1999. While at the same department she has continued to perform basic phonetic research on, e.g., vowel perception and the phonetic properties of informal Finnish dialogue speech, work that has been reported in 13 publications. Her future doctoral dissertation will concentrate on similar topics. At the University of Helsinki she teaches regular courses in acoustic-phonetic methodology. She has also worked in several speech technology related projects and authored a guide for the multi-layered annotation of speech corpora. E-mail: mietta.lennes@helsinki.fi, WWW: http://www.helsinki.fi/~lennes

MULTILINGUAL LEXICON DESIGN TOOL AND DATABASE MANAGEMENT SYSTEM FOR MT

Gintaras Barisevicius*, Bronius Tamulynas**
*Dept. of Software Engineering, Kaunas University of Technology (KTU), Lithuania,
**Computer Networking Dept., KTU, Lithuania

Abstract

The paper presents the design and development of English-Lithuanian-English dictionary-lexicon tool and lexicon database management system for MT. The system is oriented to support two main requirements: to be open to the user and to describe much more attributes of speech parts as a regular dictionary that are required for the MT. Programming language Java and database management system MySql is used to implement the designing tool and lexicon database respectively. This solution allows easily deploying this system in the Internet. The system is able to run on various OS such as: Windows, Linux, Mac and other OS where Java Virtual Machine is supported. Since the modern lexicon database managing system is used, it is not a problem accessing the same database for several users.

Keywords: lexicon, computer-based translation, Mysql, database managing

1. Introduction

The design and development of English-Lithuanian-English (ELE) dictionary-lexicon tool and lexicon database management system (DMS) for MT is oriented to support two main requirements: to be open to the user and to describe much more attributes of speech parts as regular dictionary that are required for the MT. Currently, the lexicon is designed to support all parts of speech for Lithuanian and English languages. It is possible to extend the same database for other pairs of languages as well. Polysemy problem is overcome adding an additional table between two tables linking different translations of the word in the target language. The translations for the same words are enumerated in descending priority in both directions. In this way it is possible to ensure that even if the translation won't be very exact, the user will be able to choose the suitable words himself. The possibility to include additional context attributes for the nouns is allowed. Thus the meaning of the word to be translated first will be chose according to the context and highest priority. The user friendly interface is quite comfortable in that way that it is possible to see all generated forms and that is very efficient in the process of filling dictionary with new words. There is ambition to use some additional interface features that might improve user work with MT system: such as other meaning choice or selection of words that are being translated and etc. So, if the translation won't be very exact, the user will be able to define the suitable words adequacy himself. Java and database management system MySql is used to implement the design tool and lexicon database respectively. This solution allows easily to be deployed the MT system in the Internet. The system is available to run on various OS such as: Windows, Linux, Mac and other OS where Java VM is supported. Currently, only one lexicographer can work on the system simultaneously, but the import (merging two dictionaries into one) function is under way. Since we are using modern DMS, it is not a problem accessing the same database for several users and mutual exclusion principle is handled by database managing system itself. The Lithuanian Government approved to support this project according to the national program "Lithuanian language in Information society for the years 2005-2006 for the development of the Lithuanian language technologies including computer-based translation".

2. The linguistic databases and electronic dictionaries for MT: problems and solutions

For a long time there was only one bidirectional electronic dictionary WinLED in Lithuania[1] for Lithuanian-English and Lithuanian-German language pairs. The dictionary was comfortable to use, since it had a „copy and translate" function, which allows automatically translate the text copied to the OS clipboard. Nevertheless, WinLED has several disadvantages: the user interface is in English only, the wordlist is too short and misleading translation can be occurring.

Just recently a new dictionary „Tildės biuras" has been distributed.[2] It also maintains bidirectional Lithuanian-English and Lithuanian-German dictionaries. English-Lithuanian dictionary contains 50,000 words and phrases. The dictionary proposes not only the translations, but word pronunciation, part of speech and usage of the word. The Lithuanian user interface is quite important since a major part of Lithuanian population is barred from e-content due the lack of good English skills.

An electronic version of B. Piesarskas „Didysis anglų-lietuvių kalbos žodynas" [Great dictionary of English-Lithuanian languages] „Alkonas"[3] is more comfortable than previous dictionaries, since it contains the book format and has a huge word list (around 100,000) with the words usage examples. The translation is bidirectional, but the problem is that words are indexed only from English to Lithuanian language and Lithuanian word translation search is performed using the same index.

All these dictionaries are only alternatives for the paper dictionaries since they are more comfortable and are possessed by faster search engine. Nevertheless, they have no additional data that is required for the MT, such as declensions, conjugations and etc. These available electronic dictionaries are generally different from those that are needed for the MT system. The problem is concerned with the requirement to perform not only word-to-word translation but to adjust the syntactic and semantic information as well as further processing of the target text.

The 100 million word text corpus for Lithuanian language[4] is a very important resource for building new lexicons for MT. There are still no operational parallel corpora for English and Lithuanian languages. Naturally, the building such corpora is time and effort consuming, since its designing process is rather complex. As an alternative for parallel corpora could be comparable corpora. Such corpora should

[1] VteX company http://www.led.lt/

[2] http://www.tilde.lt/ biuras

[3] http://www.fotonija.lt

[4] http://donelaitis.vdu.lt

contain texts in both English and Lithuanian languages and disseminated according to the topics of the texts.

The research on Computer-Based Translation (CBT) was initiated at KTU in 2001 (Tamulynas 2004: 16–19). Several prototypes of CBT systems have been produced, but every time the same problems have been encountered: they were robust and incomplete in terms of lexical data (Misevičius et al. 2002: 38–45).

3. Conceptual linguistic structure of lexicological database model

Lexicon for MT (Hutchins 1992 or Trujillo 1999) is a main and most important part of the system. Nevertheless there is no theoretical background what conceptual structure of lexicological database model should prevail (Melamed 1998). The fundamental requirements connected with the functionality of CBT system are: flexibility, correctness, translation quality and etc.

According to the Lithuanian grammar (Ambrazas et al 1997) it seems enough to keep the word stem in the dictionary only. Actually, this information is not sufficient. We must keep the conjugations, genders, declension, numbers and other attributes. Suitable forms of the Lithuanian words can be generated according to the Lithuanian grammar rules. The stem then is concatenated with the endings (sometimes the stem also has to be modified).

Prepositions take a very important role in the text. They usually determine the case of the word in the text. So, they must be stored with reference what case do they require. The translation of preposition combinations (phrases) will be incorporated into the phrase translation. The conceptual structure of ELE dictionary (lexicon) consists of:

- English words with grammatical and semantic attributes as well as with additional references to the roles they can perform in the sentence, when they belong to one or another part of speech, e.g. the noun can be object, subject and etc.;
- corresponding Lithuanian words with all grammatical and semantic attributes.

The semantic information about the correspondence in English and Lithuanian languages should also be included and possible translation variations might be interpreted. The semantic information consists of the rules or prescriptions how the words may be used in certain context and etc.

Previous CBT prototypes (Misevičius et al. 2002: 38–45) did not support all parts of speech and there was lacking some morphological forms of those parts of speech that have been implemented. Noteworthy, that this new ELE version overtakes all English and Lithuanian parts of speech including noun, verb, adjective, pronoun, numeral etc. A large number of all generated morphological forms (see Table 1) are displayed in tables. In this case it is very convenient to compare the complexity, variety and to check differences between languages. For example, the variety (Ambrazas et al. 1997) of Lithuanian regular noun has 14 forms, adjective 147 forms and verb over than 229 (around 400). To enter all those words manually, would be time and effort consuming, so automating this process as much as possible is a fair solution. The ending is the variable part of word, which shows the relation with other words in the sentence.

The meaning of word is derived according to its possible usage, relation with other parts of speech and its place in the sentence, context and etc. The predicate (verb) is most important since it determines how other words are situated in the sentence and how can they be linked (context dependency).

The structure of the dictionary is oriented to the objects. Thus ELE lexicon DB is open and operates virtually, i.e. there is a possibility to add new words, terms, phrases

Table 1. Grammatical categories of Lithuanian and English

Part of speech	English	Lithuanian
Noun	▪ Feminine and masculine genders. ▪ Number: the singular, the plural; singular nouns (*milk*), plural nouns (*scissors*). ▪ The genitive case (*boy – boy's*).	▪ Feminine and masculine genders. ▪ Number: the singular, the plural; singular nouns (*laimė*), plural nouns (*žirklės, akiniai*). ▪ Case is of great importance: it indicates a relationship of a word with other words in a collocation and the sentence. There are 6 cases. There are 5 declensions.
Adjective	▪ Making comparisons: positive degree (*good*), comparative degree (*better*), superlative degree (*best*).	▪ Number &case have to match together (*gerų žmonių,...*). ▪ 2 genders: masculine&feminine. ▪ Declensions: separate masculine and feminine. ▪ Relative (non-comparable). ▪ Qualitative: pronominal; comparable (positive, comparative and superlative degrees).
Numeral	▪ Ordinal (*first*). ▪ Quantitative (*one*): principal, fractional.	▪ Ordinal: characterized by number, case and gender; share grammatical characteristics with adjectives, but are non-comparable. ▪ Quantitative: principal (have genders, cases and the same declensions as adjectives); plural (have genders and the same declensions as adjectives); collective; fractional.
Pronoun	▪ Personal (*I, me, you*). ▪ Demonstrative (*this*). ▪ Interrogative (*who*). ▪ Possessive (*my*). ▪ Reflexive (*myself*). ▪ Negative (*none, no-one*).	▪ Personal (used as nouns, possess cases): *aš, tu*. ▪ Demonstrative (have cases, numbers and genders): *tas, ta, to, tą*. ▪ Interrogative and indefinite pronouns possess cases: *kas, ko, …* ▪ Others types.
Verb	▪ Regular and irregular verbs. ▪ Conjugating an auxiliary verb+ standard form of the verb. ▪ Main categories of tenses are present, past and future tenses (there are 12 tenses overall). ▪ Modal and auxiliary verbs.	▪ Inflective by person. ▪ Possesses the category of mood (direct, subjunctive, imperative). ▪ 4 tenses: present, 2 past, future. ▪ Characterized by number. ▪ Forms, non-inflective by person: the infinitive, participles.
Adverb	▪ Degrees: *well, better, best.*	▪ Degrees: *gerai, geriau, geriausiai.*

or expressions. This property greatly increases the translation quality. Word dissemination to notional categories makes sentence analysis easier in case the word has several possible translations. It helps to choose one translation according to the meaning. On numerous occasions correct sentence translation is possible when the particular word meaning is determined by the grammar rules. The polysemy is realized to both translation directions using an additional table which links two tables of different parts of speech.

Since the system is planned to be developed further, it is desirable to have an additional interface features that might improve user case with CBT. The user interface should be able to help the user to choose the translation alternatives for the possible translation list, as well as to use the history of translation.

4. Multilingual lexicon design and database management tools

We have chosen very flexible way of implementing our lexicon (Barisevičius, Černys 2004). Since we define our queries to the data using SQL (Standard Query Language), so the exact DMS is not essential (we use MySql). The usage of DMS make the lexicon easy to modify: to add new attributes, delete them, or modify the names or types and etc. It is possible to extend the same database to other human languages as well. Since MySql can store up to 2Gb of data we don’t have to worry about the volumes anymore. The running test with database of 20 million words and the retrieval of the first word took 0.03 seconds, which was longer due to making the connection to the database. The retrieval time of the following searches was less than 0.01 seconds.

We considered the domain possibility for the nouns, thus user can choose the priorities of finding words in certain domain. This way user can prioritize the word translation according to his domain of interest.

While Java programming language for implementation is used it is possible to make system available on-line. Naturally, there is a small disadvantage of Java that it is quite slow, but the recent releases of Java Runtime Environment had improved their performance a lot. Shrinking the compiled code is another option to increase the performance. As the names of variables are shrunk to be minimal, the system operates with shorter names. Besides, the system is available on Windows, Linux, Mac and other OS where the Java Virtual Machine is supported.

The user friendly interface is designed to optimize the work of the lexicographer, that he could select the attributes of the word and generate all possible morphological forms. This feature should save amounts of lexicographers work, since he won‘t have to enter each form individually and besides will be able to see all generated forms in the screen in a very compact way.

Another problem of filling is for the lexicographers to work simultaneously. This problem can be easily solved if the system is running on-line. Since, we use state-of-art DMS, we can access the data simultaneously. All the mutual exclusion problems are handled by the database system and we don’t have to worry about that. The database export and import functions are under way. Till the end of the year, we are planning to fill the dictionary with around 20,000 of words. That should be enough for primitive translation using the most frequent words in the language.

5. Conclusions

English-Lithuanian-English dictionary-lexicon tool and lexicon database management system for MT is oriented to support two main requirements: to be open to the user and to describe much more attributes of speech parts as a regular dictionary that are required for the MT. Programming language Java and database management system MySql is used to implement the designing tool and lexicon database respectively. This solution allows easily deploying this system in the Internet. The system is able to run on various OS where Java Virtual Machine is supported. Currently, only one lexicographer can work on the system simultaneously, but the import (merging two dictionaries into one) function is under way. The research is performed according to the national program "Lithuanian language in Information society for the years 2005-2006 for the development of the Lithuanian language technologies including computer-based translation". The Tool was demonstrated in DWS (Dictionary Writing Systems workshop, Brno) in September 6-7, 2004, as well as in workshop took place in Vytautas Magnus University in October 26, 2004.

References

Tamulynas, Bronius 2004. Language processing and Multilingual communication: Lithuanian case. In: Proceedings of the first Baltic Conf. *Human language technologies. The Baltic perspective*, Riga, April 21-22, 16–19.

Misevičius Gediminas; Tamulynas B.; Žemaitis M. 2002. Anglų kalbos tekstų kompiuterinio vertimo į lietuvių kalbą technologijos. In: Liubinienė, V. (eds.) *Kalbų studijos 2*, 38–45.

Hutchins William J.; Somers H. L. 1992. An Introduction to Machine Translation. London: Academic Press.

Trujillo, Arturo 1999. Translation Engines: Techniques for Machine Translation. London: Springer.

Melamed, Dan I. 1998. Empirical Methods for MT Lexicon Construction. In: Gerber L.; Farwell D. (eds.) *Machine Translation and the Information Soup.* London: Springer-Verlag.

Ambrazas, Vytautas 1997 (ed.). Dabartinės lietuvių kalbos gramatika, Vilnius: Mokslo ir enciklopedijų leidykla.

Barisevičius, Gintaras; Černys Elvinas 2004. English-Lithuanian and Lithuanian-English Lexicon database management system for MT. In: Proceedings 3d International Workshop on *Dictionary Writing Systems (DWS 2004)*, Czech Republic, Brno: (Demonstration).

BRONIUS TAMULYNAS helds a position of Assoc. prof. in the dept. of computer networking, Kaunas University of Technology. He received his Ph.D, dealing with intelligent simulation of complex decision making systems. He has improved his professional experience in Manchester Metropolitan and Nottingham universities. Currently, he is specializing in the implementation of new information technologies in education and is responsible for dissemination of GRID networking services under FR6 IST educational project. Research areas: intelligent information technologies, semantic-based knowledge retrieval and computer-based translation. E-mail: bronius.tamulynas@ktu.lt

GINTARAS BARISEVIČIUS is a MA student of software engineering at the Kaunas University of Technology. He has studied in Link ping University, Sweden (practical skills in programming and software engineering in general). He has participated in the Nordic Graduate School of Language Technology. Research areas: machine translation, text corpora. E-mail: gintaras.barisevicius@stud.ktu.lt

DIALOGUE ACTS AND COMMUNICATIVE STRATEGIES IN ESTONIAN DIALOGUES

Liina Eskor
University of Tartu (Estonia)

Abstract

Communicative strategies are an opportunity to annotate dialogues both in human communication and human-computer interaction. I have annotated a collection of dialogues from the Estonian dialogue corpus (EDiC) using Kristiina Jokinen's theory of communicative strategies, compared two annotating systems of EDiC – dialogue acts and communicative strategies – and analyzed if there are any fixed accordances between the acts and strategies.

Keywords: dialogue corpus, spoken human-human dialogues, dialogue annotation, dialogue acts, communicative strategies

1. Introduction

The present study is a part of a project the goal of which is to build a dialogue system that would be able to interact with humans in Estonian using the norms and rules of human-human communication.

The study is based on dialogues from the Estonian dialogue corpus that consists of both human-human and simulated human-computer dialogues.

The corpus includes 623 spoken human-human dialogues (transcribed according to the notation of conversational analysis), including 508 calls for information and 115 face-to-face conversations, with the total length of 100 000 running words.

There are also 21 dialogues (about 2500 running words) in EDiC that were collected during computer simulations using the Wizard of Oz (WOZ) method (Valdisoo et al. 2003). A group of 11 persons was asked to test a program that would interact in Estonian and provide travel information but actually a human (Maret Valdisoo) played the role of the computer.

The corpus is annotated according to a typology of dialogue acts (Gerassimenko et al. 2004). The typology departs from the point of view of conversational analysis that focuses on the techniques used by people when they are actually engaged in social interaction. The total number of dialogue acts is 126. The acts are divided into two big groups – adjacency pair acts (e.g. directives and grants of them) and single acts (e.g. acknowledgement). The names of the acts consist of two components: the first component gives information about the general type of the act (e.g. DIF – directives, the first part; DIS – directives, the second part; RE – responses). The second component shows the concrete name of the act (e.g. DIF: request, DIS: giving information, RE: acknowledgement).

In my study (Eskor 2004) I analyzed 40 dialogues from EDiC: 20 calls for information and 20 Wizard-of-Oz dialogues (WOZ dialogues).

2. The theory of communicative strategies

In addition to dialogue acts I used another possibility to annotate dialogues: communicative strategies. The concept used in this study comes from Kristiina Jokinen's Constructive Dialogue Management (CDM) approach. The model originates from the general communicative principles that constrain cooperative and coherent communication. With the help of these principles, the dialogue system can reason about appropriate communicative strategies and overcome shortcomings in its knowledge base (Jokinen 1996, 1998). The reasoning about an appropriate strategy means that the system has to choose between different possibilities how to carry on a conversation (ask for additional information, advise other possibilities, request to repeat the question, etc). To put it in a more formal way – communicative strategies are ways in which mutual knowledge is established, maintained, modified and exploited. Communicative strategies are based on the agent's rationality: actions are chosen so as to conform to the shared assumptions about cooperative behaviour in a given situation.

A communicative strategy is used by a participant to build up the next turn as a reaction to the partner's previous one. Communicative strategies, therefore, express the coherence of the dialogue like adjacency pairs of dialogue acts.

Some strategies that the agents use to process mutual information turn out to be more successful than others. Thus, it is not enough to provide only the factual information. It is also important to note that the better the agents solve the emerging problems during a conversation, the more accurately and quickly the goal of the interaction is achieved.

The communicative strategy is chosen on the basis of four contextual factors (Jokinen 1996):

1. **Expectations**: whether or not the partner's contribution conforms to the expectations evoked by the speaker's previous contribution.

2. **Central Concept**: whether or not the partner's contribution is thematically related to the previous central concept.

3. **Initiatives**: whether or not the speaker has the initiative.

4. **Goals**: whether or not the speaker has unfulfilled goals.

All the context factors have binary values in CDM, which results in 2^4=16 communicative strategies.

A stack is an appropriate data structure to describe the setting up and abandoning of goals during a dialogue where the LIFO (last in, first out) principle holds (Eskor 2002). The client's first question sets up the main goal which is put at the bottom of the stack. The following adjusting questions set up new goals which are added to the stack step by step. To achieve the main goal, all the goals in the stack that are located higher than the main goal must be achieved and removed. If all the goals are achieved then the stack is empty. Such a situation holds at the beginning of a conversation (before the first request of the client) and at the end of conversation if the client has received answers to all the questions.

It turns out that for determining the right strategy it is very important to consider the roles of the speaker and the partner. In the analyzed information request dialogues it was very common that the initiative was transferable: depending on the situation, both agents could take the initiative.

When comparing communication strategies in two different types of dialogues it turns out that the most common strategy is *follow-up-old* (45,9% of all strategies in WOZ dialogues, 32,1% in spoken dialogues). Other regular strategies are also *finish/start* (15,7% in WOZ dialogues, 16,7% in spoken dialogues) and *subquestion, X* (11,2% in WOZ dialogues, 7,7% in spoken dialogues; Figures 1 and 2).

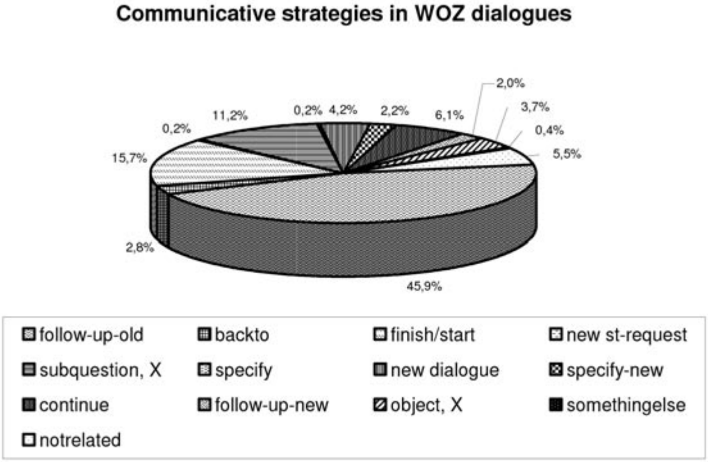


Figure 1. Communicative strategies in WOZ dialogues

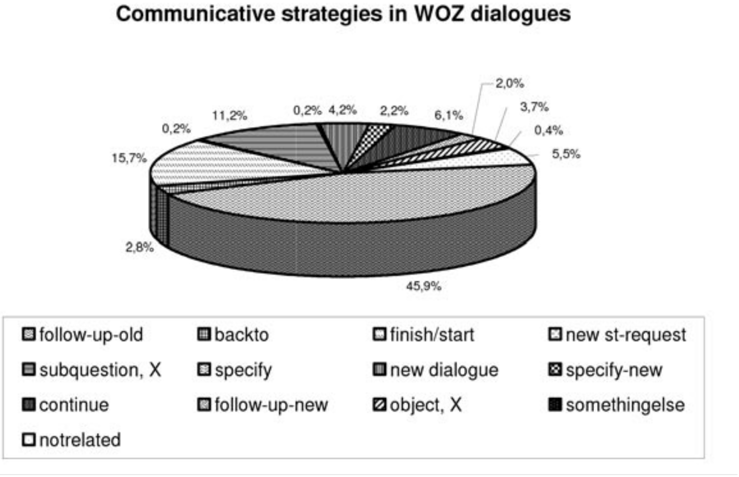


Figure 2. Communicative strategies in spoken human-human dialogues

3. The structure of dialogues

Having established that two different types of dialogues (spoken and WOZ dialogues) were used in different communication channels, it turned out that the dialogues have a bit different structures. WOZ dialogues tend not to have any conventional beginning (greeting) and the agent starts to ask questions at once. At the same time it is very typical to talk about many different topics during one conversation. The reason is that the wizard prolongs conversations intentionally, it always proposes continuing the dialogue. This does not usually occur in spoken dialogues. Spoken dialogues have more complex structure, involving feedback from the partner (such as continuers, acknowledgements) and long conventional greetings and closings.

4. Relations between dialogue acts and communicative strategies

Next I will give an overview of the most common relations between dialogue acts and communicative strategies. The observed acts and strategies are in italics in the examples provided.

Finish/start is related to *DIF: request* or *QUF: wh-question*. *Finish/start* always marks the first question about a new topic. It can be at the beginning of the conversation as well somewhere in the middle. It is important is that the stack is empty, i.e. there are no unfulfilled goals at the moment.

Utterance	Dialogue act	Strategy	Goal stack
Infoklient: Kuidas on kõige kiiremini võimalik jõuda Tartust Kuressaarde? Client: what is the fastest way from Tartu to Kuressaare?	*QUF: wh-question*	*Finish/start*	
			Client's question

Subquestion, X and *QUF: adjusting the conditions of answer* mark the wizards' or information officers' request to specify the question (e.g. the starting point or the destination of the trip). That kind of annotation pair is very common in the WOZ dialogues.

Utterance	Dialogue act	Strategy	Goal stack
Infoklient: Kas Sebe bussid Saaremaal liiguvad? Client: Does Sebe have busses in Saaremaa?	QUF: open yes/no question	Finish/start	
Arvuti: Täpsustage sihtpeatuse nimi, palun! Computer: Specify the destination, please!	*QUF: adjusting the conditions of answer*	*Subquestion, X*	Client's question
			Adjusting/ repair Client's question

Follow-up-old and *QUS: giving information* mark both the answers of the client and wizard/officer which provide the expected information.

Utterance	Dialogue act	Strategy	Goal stack
Infoklient: Kuidas sõita Tartust Pärnu enne kella 12 hommikul Client: How to get to Pärnu from Tartu before 12 in the morning	QUF: open yes/no question	Finish/start	

Arvuti: buss väljub kell 05.00 Computer: The bus departs at 05.00	*QUS: giving information*	*Follow-up-old*	Client's question
Arvuti: buss väljub kell 05.00 Computer: The bus departs at 05.00	*QUS: giving information*	*Follow-up-old*	Client's question
			~~Client's question~~

Continue and *RE: continuer* appear together only in spoken dialogues. The speaker interrupts the partner just for a moment to confirm that he is listening. After that the partner continues giving the answer. So the continuer is just a part of a fluent conversation[1].

Utterance	Dialogue act	Strategy	Goal stack
H: e tere ma paluksin (.) alates kella 'kuuest Tallinna 'ekspressi 'aegasid Client: I'd like to get the bus timetable to Tallinn from 6 o'clock	RIF: greeting DIF: request	- Finish/start	
V: m seitseteist='viiskümmend Officer: Seventeen fifty	DIS: giving information	Follow-up-old	Client's question
H: jah Client: yes	*RE: continuer*	*Continue*	Client's question
V: kaheksateist='kakskümend kiirliin Officer: Eighteen twenty express	DIS: giving information	Follow-up-old	Client's question
			~~Client's question~~

Continue and *DIS: missing information* indicate the wizards'/officers' utterance. It means that there in no opportunity to provide the information. This situation is quite common both in spoken and WOZ dialogues.

Utterance	Dialogue act	Strategy	Goal stack
Arvuti: Kas Teid huvitab mingi konkreetne nädalapäev? Computer: Are you interested in some specific day of the week?	QUF: open yes/no question, QUF: adjusting the conditions of answer	Subquestion, X	Client's question
Infoklient: reede Client: Friday	QUS: giving information	Follow-up-old	Adjusting/ repair Client's question
Arvuti: Reedel lennuliiklust Tartu ja Helsinki vahel pole! Computer: There is no air traffic between Tartu and Helsinki on Friday!	*DIS: missing information*	*Continue*	~~Adjusting/ repair~~ Client's question
			~~Client's question~~

Here we can also see that the same strategy (*continue*) corresponds to different dialogue acts in different situations (here: *continuer* and *missing information*).

[1] Transcription of conversational analysis is used in spoken dialogues.

5. Conclusion

The analysis of dialogue acts and communicative strategies confirmed the initial hypothesis that there exist fixed accordances between the acts and strategies, e.g. the strategy *finish/start* responds to different questions and requests.

The study is an important part of developing EDiC further. It was necessary to analyze whether the two-layer dialogue tagging simplifies the composing of the dialogue system. It can be confirmed that both dialogue acts and communicative strategies are necessary layers: the multiplicity of acts is necessary to find initial parts and detailed structure of dialogues, and the limited number of strategies would help the computer to understand the user questions and generate the answers.

Acknowledgement

This work is supported by the Estonian Science Foundation (grant 5685).

References

Gerassimenko, Olga; Hennoste, Tiit; Koit, Mare; Rääbis, Andriela; Strandson, Krista; Valdisoo, Maret; Vutt, Evely 2004. Annotated dialogue corpus as a language resource: an experience of building the Estonian dialogue corpus. In: *The first Baltic conference "Human language technologies. The Baltic perspective". Commission of the official language at the chancellery of the president of Latvia*. Riga. 150 – 155.

Eskor, Liina 2002. Dialoogiaktid "võlur Ozi" tehnikaga kogutud dialoogides. Bakalaureusetöö, Tartu Ülikool.

Eskor, Liina 2004. Dialoogiaktid ja suhtlusstrateegiad: eesti dialoogikorpuse analüüs. Magistritöö, Tartu Ülikool.

Jokinen, Kristiina 1996. Cooperative Response Planning in CDM: reasoning about communicative strategies. In *Proceedings of the 11th Twente Workshop on Language Technology: Dialogue Management in Natural Language Processing Systems*. Twente: the Netherlands. 159-168.

Jokinen, Kristiina 1998. Three challenges for dialogue management: the constructive dialogue model approach. In Haukioja, T. (ed.) *Papers from the 16th Scandinavian Conference of Linguistics*, Turku, Finland. 221 – 234.

Valdisoo, Maret; Vutt, Evely; Koit; Mare 2003. On a method for designing a dialogue system and the experience of its application. In *Journal of Computer and Systems Sciences International* 42(3). 456-464.

LIINA ESKOR is Ph.D. student in general linguistics, University of Tartu. She received her MA (general linguistics) from the University of Tartu in 2004. Her research interests are spoken language and dialogue annotation. Her doctoral study focuses on finding the adequate definition of a communicative strategy, discovering and describing the strategies in Estonian spoken dialogues, analyzing their implementation in different situations and building a formal model that describes the structure of dialogues. E-mail: liina.eskor@mail.ee.

DIALOGUE ACT RECOGNITION IN ESTONIAN DIALOGUES USING ARTIFICIAL NEURAL NETWORKS

Mark Fishel
University of Tartu (Estonia)

Abstract

This paper describes two experiments of applying dialogue act recognition to Estonian dialogues. Two class systems were used in both of them — a general one (with 19 classes) and a detailed one (with 107 classes).

In the first experiment the task was performed using learning vector quantization (LVQ). The preprocessing was done in WEBSOM (Self-Organizing Maps for Internet Exploration) style; the used method was originally designed for processing documents with self-organizing maps, with which LVQ shares its principle. The weighing method was `tf×idf`, which is the most popular method for term weight assignment.

The first experiment wasn't a success. The proposed explanation is that due to the difference between utterances and text documents the combination of LVQ and the used preprocessing method were too straightforward for this task, and a more sofisticated classifier was to be tested.

In the second experiment multilayer perceptron (MLP) techniques were applied to data, preprocessed in the same way as in the first experiment. The tested networks had one and two hidden layers. The experiment ended successfully; networks with two hidden layers had shown the highest accuracies with both class systems.

Keywords: dialogue act recognition, learning vector quantization, multilayer perceptron

1. Introduction

The purpose of this work is to study application of neural network classification to the Estonian language. The exact task is to apply the technique to dialogue act recognition.

A dialogue act is basically a class of dialogue utterances of the same type (*statement*, *question*, *greeting*, etc). To recognize dialogue acts means to label each utterance with its type (to classify).

The task of automatically recognizing dialogue acts is an important link in understanding natural language. For instance if the type of an utterance is known a conversational agent can respond to it appropriately. A more detailed description of the possible applications of the task can be found in (Stolcke et al. 2000).

As far as the author knows this is the first attempt of performing automatic dialogue act recognition with Estonian dialogues.

2. Experiments

The data used for training and testing was gathered during the Estonian Dialogue Corpus (EDiC) project (Gerassimenko et al. 2004). In total there are 113 dialogues containing 5624 utterances; the dialogue acts were attributed manually.

The classes and the dialogue act system used in this work have been introduced in the same project. A class name consists of an abbreviation of 2 or 3 letters indicating the general type, and a full description; for example *RIE TERVITUS* (conventional, the first part: greeting), *KYJ JAH* (questions, the second part: yes). Two classification sets are used — the short name set $\mathcal{C}_{\mathcal{S}}$ (with only the abbreviation taken into consideration) and the full name set $\mathcal{C}_{\mathcal{F}}$ (which uses the whole class name). The first set contains 19 classes, the second – 107. More information on the dialogues and the class system (including the list of class names and their meanings) can be found in (Gerassimenko et al. 2004).

In both experiments 25% of the whole data set is used for training and the remaining 75% for testing, producing two statistically independent sets. This separation quota is based on (Ruiz and Srinivasan 2005).

2.1. Preprocessing

The initial idea was to repeat the experiment described in (Jokinen et al. 2001) with the Estonian language — namely to recognize dialogue acts by using learning vector quantization.

Utterances are preprocessed in WEBSOM-style (Jokinen et al. 2001). All the utterance words are stemmed using the morphological analysis software, obtained from the Institute of the Estonian Language (Morfoloogiline analüüs). Each stem is then associated with a random vector. Vector size of 90 has been shown to be sufficient (Honkela 1997) but vectors of smaller sizes (10 and 40) were also tested in this work. The vector representing an utterance is obtained by calculating the weighed sum of all the vectors of the words in it; the weighing principle used is `tf×idf` (term frequency/inverse document frequency).

It turned out that text representation used in (Jokinen et al. 2001) contains a methodological error. Namely, in `tf×idf` weighing the term frequency is taken relative to the class the utterance belongs to. This way some information about the output is used to encode the input which naturally gives a significant boost to recognition precision, but also makes it impossible to encode an unclassified utterance; therefore the method cannot be used in practice.

The solution to this problem chosen by the author of this work was to take term frequency relative to the utternace itself. This complies to the original approach of WEBSOM-style preprocessing but isn't bound to work in our case because of the difference between a dialogue utterance and a text document. An utterance, on the average, as short as five to ten words and it is quite uncommon for a word to be present more than once in it; therefore the term frequency component in `tf×idf` equals 1 in most cases. Nevertheless, the method is still being tested.

2.2. Applied Techniques

The applied techniques were learning vector quantization (LVQ) and multilayer perceptrons (MLP). In comparison with MLP, LVQ is simpler and somewhat more straightforward in principle; its advantage, however, is that the preprocessing method that is used

Table 1: The resulting accuracies

class set	$\mathcal{C_S}$	$\mathcal{C_F}$
LVQ accuracy	39%	27%
MLP accuracy	83%	63%

here was designed especially for self-organizing maps (SOM) with which LVQ is very similar (although the purposes of SOM and LVQ differ, their principles are almost identical). MLP on the other hand can find more complex dependencies, but may take longer to converge while training (not to mention that it doesn't converge stably).

As an implementation of the LVQ the author uses `lvq_pak` (SOM_PAK and LVQ_PAK). The training algorithm is `olvq1` with 200,000 iterations, the starting learning rate parameter α = 0.045; the parameter values are chosen empirically. The number of codebook vectors is 2,000.

The software used for the MLP classification is Stuttgart Neural Network Simulator (SNNS). The tested networks are with one and with two hidden layers.

3. Results

The resulting classifier precisions are shown in Table 1; it can be clearly seen that the LVQ classification failed. However, it is interesting to notice that some dialogue acts are recognized with high precision, whereas they are not the most frequent ones — for instance *VR* (responces) in $\mathcal{C_S}$ or *RIE TÄNAN* (conventional, the first part: thanking) in $\mathcal{C_F}$. The reason for that most probably lies in the fact that utterances in these dialogue acts are with limited lexicon and usually have 1 or 2 words, thus allowing LVQ to find the few variants precisely while learning.

MLP showed significantly better accuracy than LVQ; the parameters of the networks that showed best results are shown in Table 2. In both cases there were two hidden layers — apparently, one layer is not enough to solve this task efficiently. Also, it can be seen by the number of input neurons that smaller input vector sizes are less efficient for this task.

In case of MLP higher accuracy follows classes with more entries while infrequent class accuracy is sometimes as low as 0%. Knowing the nature of MLP this is understandable — the classes presented scarcely seem more like exceptions to the classifier and due to its generalizing ability they do not influence the system very much.

In the author's opinion, the only possible explanation for the poor performance of LVQ is the inapplicability of its combination with the WEBSOM-style preprocessing to dialogue act recognition. To be more exact, the problem with vector weights mentioned in experiment descriptions causes different vector subspaces, each corresponding to a different class, to be mixed and therefore confused by the LVQ classifier. A possible solution to the problem would be either to choose a different weighing model or to use a more complex learning system. The second variant proves to be successful in case of multilayer perceptrons, as shown by the second experiment.

Table 2: “Winner” multilayer perceptron parameters

class set	input neurons	hidden neurons	η	α
$\mathcal{C}_{\mathcal{S}}$	90	45	0.05	0.6
$\mathcal{C}_{\mathcal{F}}$	90	60	0.02	0.4

4. Future Work

The author of this work is currently analyzing the results of experiments of applying recurrent neural networks to the same task. One of the main focuses is the data preprocessing methods, not involving dictionary construction — the problem with the methods that do involve it is that only words present in training data (using which the dictionary is constructed) are recognized while the rest are simply ignored.

Some ideas that might help in dialogue act recognition have been left unapplied in this work. One of them is that consequent utterance dialogue acts are not independent. For instance, it is highly probable that a question is followed by an answer. Providing the neural network with the knowledge of preceeding (already recognized) dialogue acts might boost the recognition precision.

Another idea concerns mostly the classification system taken from the EDiC project (Gerassimenko et al. 2004). In this work experiments with the short and full class names were performed independently. However this ignores the fact that the full name is a specification of a short name — i.e. if the short name of some dialogue act is *RIE*, the full name cannot be *KYJ JAH*, for it has to start with *RIE*. One of the ways to take advantage of this would be to use the output of a trained short name classifier as an additional input of a full name classifier.

References

Gerassimenko, O.; Hennoste, T.; Koit, M.; Rääbis, A.; Strandson, K.; Valdisoo, M.; Vutt, E. 2004. Annotated dialogue corpus as a language resource: An experience of building the estonian dialogue corpus. In: *Proceedings of the First Baltic Conference “Human Language Technologies. The Baltic Perspective”. Commission of the Official Language at the Chancellery of the President of Latvia, Riga.* 150–155

Honkela, T. 1997. Self-Organizing Maps in Natural Language Processing. PhD thesis, Helsinki Univ. of Technology, Espoo, Finland

Jokinen, K.; Hurtig, T.; Hynnä, K.; Kanto, K.; Kaipanen, M.; Kerminen, A. 2001. Self-organizing dialogue management. In: *Proceedings of the Natural Language Pacific Rim Symposium (NLPRS) Workshop “Neural Networks and Natural Language Processing”, Tokyo, Japan*

Morfoloogiline analüüs. Retrieved December 12, 2004, from http://www.eki.ee/tarkvara/analyys/

Ruiz, M. E.; Srinivasan, P. 2005. Automatic Text Categorization Using Neural Networks. Retrieved February 19, 2005, from http://informatics.buffalo.edu/faculty/ruiz/publications/sigcr97/sigcrfinal2.html

SNNS. Stuttgart Neural Network Simulator. Retrieved December 30, 2004, from http://www-ra.informatik.uni-tuebingen.de/SNNS/

SOM_PAK and LVQ_PAK. Retrieved December 8, 2004, from http://www.cis.hut.fi/research/som-research/nnrc-programs.shtml

Stolcke, Andreas; Ries, Klaus; Coccaro, Noah; Shriberg, Elizabeth; et al 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. In: *Computational Linguistics 26:3* 339–373

Aknowledgements

The work is supported by the Estonian Science Foundation (grant 5685).

MARK FISHEL is an undergraduate student at the University of Tartu. His research interests include language technology, neural networks and machine learning in general.

FORMAL SPECIFICATIONS FOR A DEPENDENCY GRAMMAR OF THE LITHUANIAN LANGUAGE

Gintarė Grigonytė, Erika Rimkutė
Vytautas Magnus University, Lithuania

Abstract

The first attempt to create some rules of Dependency Grammar (DG) for the Lithuanian language is introduced in this article. The need of a Lithuanian language parser was the background of this research. Concerning Lithuanian language processing, there are some key works on morphology level, but syntactical analysis is still lagging behind. That is the main reason why we consider formal specifications of DG being important for Lithuanian language processing. Up to now only verb phrases were syntactically analyzed while the grammar we propose involves a fuller coverage of the Lithuanian syntax. Our approach is based on corpus-based methods that let us extract, classify and evaluate DG rules. The main results of this research are discussed here as well.

Keywords: grammatical dependency, syntactic analysis, corpus, dependency rules, word order, insertion, word group

1. Introduction

Lithuanian language is a highly flective language. There are some important works in automatic morphology area. The first attempts of automatic syntactic analysis are introduced in this article. The proposed specification for formal syntactic analysis is based on corpus-based rules. The main reason why we have chosen this method is a lack of available works and results concerning not only the formal grammar of the Lithuanian language but also computational linguistics in general.

Our linguistic resources were semi-automatically annotated Lithuanian corpus that consists of 1 million running words; the Corpus of the Contemporary Lithuanian Language (both corpuses were created at the Centre of Computational Linguistics, Vytautas Magnus University); the morphological analyzer *Lemuoklis*. The rules, presented bellow, include the most frequent met syntactic structures.

2. The specification of the formal dependency rules

Automatic analysis of the Lithuanian syntax is based on general principles of DG. According to I. Mel'čuk (Mel'čuk 1998: 13–15), DG is more relevant to the syntactic structures of a natural language than Phrase structure grammar. DG-based syntactic analyzer should recognize which word is dominant and which one is dependent.

We set four additional parameters for the descriptions of syntactic structures. They are as follows: dependency, word order, insertion, and priority. Dependency is the necessary attribute of links between words. Word order and insertion is possible but not mandatory. Priority is essential for the next stage of syntactic analysis (e.g. parser) and we will not pay much attention to this parameter here.

We mark **dependency** as an arrow which goes form the governing to the dependent word, e.g. *word1→word2*, which means *word1* governs *word2*. Possible types of dependencies are represented in figure 1: a) one way forward dependency, *word1* governs *word2*, e.g. *einu → namo* (*I am going → home*); b) one way backward dependency, *word2* governs *word1*, e.g. *mažas ← vaikas* (*a little ← kid*); c) dual dependency, both words are on the same level, e.g. *ponas ↔ Jonas* (*Mister ↔ John*).

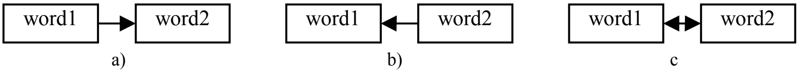


Figure 1. Possible types of dependencies between two words

Word order is the other important parameter concerned DG rules. Lithuanian language has free word order that is inconvenient for word dependency determinations. The most frequent sentence structure is SVO[1], e.g. *Jonas* (S) *skaito* (V) *knygą* (O) (*John reads a book*).

The possible word order is presented in Figure 2. It is clear that dependency here is the same but the word order is different, e.g. *Jonas skaito* (*John reads*) vs. *skaito Jonas* (literally *reads John*).

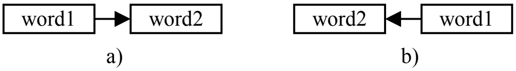


Figure 2. Possible types of word order in a two-word combination

The other important criterion for syntactic description is **insertion**. An additional word can be inserted in already established structures. Usually new inserted words do not change syntactic dependency but modify its structure. It is necessary to evaluate the parameter of the possible insertion. The main tendency is for nominal words to be inserted with other nominal and adverbial word combinations while finite verb forms are never inserted in them. Verb phrases are characterized by a great variety of forms, various possible word order and insertions.

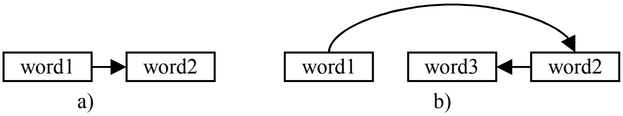


Figure 3. Example of insertion between two words

The insertion of an additional word between two words is represented in Figure 3. Insertion does not affect dependency, e.g. a) *skaito → vaikas* (*a kid → reads*); b) *skaito → [mažas ← vaikas]* (literally *reads → [little ← kid]*).

Up to now we have discussed three main relational parameters: dependency, word order and insertion. We have excluded **priority** as it is considered being the next step of

[1] S – subject, V – verb, O – object

analysis. Priority reveals the importance of different rules of the same group, e.g. adjective ← noun (*mažas* ← *vaikas* (*little* ← *kid*)), noun → adjective (*vaikas* → *mažas* (literally *a kid* → *little*)). The above mentioned rules indicate the same dependency, but the first one is more common therefore of a greater priority.

3. The dependency rules of the Lithuanian language

Grammar rules proposed here consist of two levels: the level of word groups (lower level) and the level of the combinations of word groups (upper level). We will discuss only word group level.

3.1. The dependency rules for word groups

3.1.1. Subject groups

Subject of the Lithuanian language can be simple or complex. A simple subject usually consists of one noun, e.g. *vaikas skaito* (*a kid reads; vaikas (a kid)* is a subject). An extended simple subject consists of adjectives, participles, pronouns, numerals that are compatible with the noun in gender, number and case, e.g. *maž**as** vaik**as*** (*a little kid*), *pirm**a** knyg**a*** (*the first book*) (see table 9). A complex subject consists of a few nouns in one group, e.g. *berniukai ir mergaitės* (*boys and girls*) (see table 1).

The same rules might be joined into a new complex group of rules, e.g. noun ↔ noun, [adjective ← noun] ↔ [adjective ← noun]. Various words or word groups can be inserted in a two word combinations, e.g. [adverb ← adjective] ← noun. Word order is important in that case also: adjectives, participles, numerals and pronouns usually precede nouns. A different word order is supposed to be inverted.

Table 1. The most frequent rules for a subject

Example	Word1	Word2	Word3	Dependency
Jonas Jonaitis	proper noun (G, N, C)[2]	proper noun (G, N, C)	-	proper noun ↔ proper noun*[3]
ponas Jonas	noun (G, N, C)	proper noun (G, N, C)	-	noun ↔ proper noun*
gydytojas chirurgas	noun (G, N, C)	noun (G, N, C)	-	noun ↔ noun*
P. Jonaitis	abbreviation	noun/proper noun	-	abbreviation ↔ proper noun*
Berniukai ir mergaitės	noun/proper noun	conjunction	noun/proper noun	noun/proper noun → conjunction → noun/proper noun

3.1.2. Predicate groups

Predicate can be simple and complex as well as subject. Finite verbs and participles compose simple predicates, e.g. *lyja* (*it's raining*). The most common structures of complex predicates are represented in table 2.

Table 2. The most frequent predicates rules

Example	Word1	Word2	Word3	Word4	Dependency
turėtų būti geras	finite verb	infinitive	participle/ adjective	-	finite verb → infinitive → participle/adjective*

[2] G – gender, N – number, C – case, P – person

[3] „*“ means additional parameters (word order, insertion, etc.) for dependency rules exist.

turiu eiti	finite verb	infinitive	-	-	finite verb → infinitive*
buvo priverstas dirbti	finite verb (*būti*)	participle	infinitive	-	finite verb → participle → infinitive*
turi būti verčiamas dirbti	finite verb	infinitive (*būti*)	participle	infinitive	finite verb → infinitive → participle → infinitive*
galima dirbti	participle/adjective	infinitive	-	-	participle/adjective → infinitive*
buvo einantis	finite verb (*būti*)	participle	-	-	finite verb → participle*
galima būtų pasakyti	participle	subjunctive verb (*būti*)	infinitive	-	subjunctive verb → participle → infinitive*
būtų buvę galima padaryti	subjunctive verb (*būti*)	participle (*būti*)	participle	infinitive	subjunctive → participle (*buvę*) → participle → infinitive*
norėtųsi pailsėti	subjunctive verb	infinitive	-	-	subjunctive verb → infinitive*
būtų buvęs daromas	subjunctive verb	participle	participle	-	subjunctive verb → participle (*buvęs*) → participle*
noriu eiti miegoti	finite verb	infinitive	infinitive	-	finite verb → infinitive → infinitive*
norint padaryti	gerund	infinitive	-	-	gerund → infinitive*
bėgte nubėgo	second infinitive	finite verb	-	-	second infinitive ← finite verb*
bėgte nubėgti	second infinitive	infinitive	-	-	second infinitive ← infinitive*
gyventi yra gera	infinitive	finite verb (*būti*)	adjective/ participle	-	finite verb ← [adjective/participle → infinitive]*
skaito rašo	finite verb	finite verb	-	-	finite verb ↔ finite verb*
skaito ir rašo	finite verb	conjunction	finite verb	-	finite verb → conjunction → finite verb*
atrodo sveikas	finite verb/infinitive (N)	adjective/participle/noun (N, G)	-	-	finite verb/ infinitive → adjective/participle*

3.1.3. Attribute groups

Adjectives, participles, some pronouns and numerals compose attribute groups. Usually they are coordinated with noun in gender, number and case. Attributes that are not in concord with nouns are common in Lithuanian language. They are represented by Genitive of the noun, e.g. *tėvo knyga* (*father's book*). The main rules of attribute groups are given in Table 3.

Table 3. The most frequent attributive rules

Example	Word1	Word2	Dependency
gerai žinomas	adverb	adjective/participle	adverb ← adjective/ participle*
turtingas pinigų	adjective	noun Gen[4]	adjective → noun Gen*
gabus muzikai	adjective	noun Dat	adjective → noun Dat*

[4] Gen – Genitive, Acc – Accusative, Instr – Instrumental, Loc – Locative

garsus pasiekimais	adjective	noun Instr	adjective → noun Instr*

3.1.4. Object groups

Object groups are mainly expressed by nouns in Genitive, Dative, Accusative and Instrumental cases, e.g. *skaito knygą* (*read/reads a book*), *didžiuojasi sūnumi* (*is/are proud of a son*). There are some variants when an object group consists of a preposition and a noun in Genitive, Dative, Accusative and Instrumental cases. These variants have a fixed word order and the governing word is usually a preposition. Attributes of nouns can be inserted in object groups (see Table 4).

Table 4. The most frequent object groups

Example	Word1	Word2	Dependency
ant dėžės	preposition	noun/adjective/pronoun/ numeral/participle Gen	preposition → noun/adjective/ pronoun/ numeral/participle Gen*
apie vaikus	preposition	noun/adjective/pronoun/ numeral/participle Acc	preposition → noun/adjective/ pronoun/ numeral/participle Acc*
su draugais	preposition	noun/adjective/pronoun/ numeral/participle Instr	preposition → noun/adjective/pronoun/ numeral/participle Instr*

3.1.5. Modifier groups

Adverbs, half-participles, nouns in Instrumental, Locative cases compose modifier groups, e.g. *dirbti miške* (*to work in a forest*), *greitai bėgti* (*to run fast*). Prepositional constructions with nouns can also form a modifier group. The most frequent rules for modifier rules and their internal relations are shown in Table 5.

Table 5. The most frequent modifier groups

Example	Word1	Word2	Dependency
be galo daug	adverb	adverb	adverb ← adverb
per greitai	particle	adverb	particle → adverb
prie miško	preposition	noun Gen, Acc, Instr	preposition → noun Gen, Acc, Instr*

Rules that describe main parts of sentence belong to the upper (sentence) level. Joining rules of both levels enables us fully to describe syntactic relations of a simple sentence. Upper level rules can be classified into predicate and subject relation, predicate and object relation, predicate and modifier relation, subject and attribute relation according to their sentential functions. Due to the lack of space a more detail description of these rules are omitted.

Some of the lower level rules are applied in analyzing complex verb groups (Grigonytė 2004; Grigonytė et al. 2005). The future research will go in the direction of syntactic analyzer. Another application for the automatic syntactic analysis is morphological disambiguation of Lithuanian language (Rimkutė 2003).

4. The methodology of extraction of corpus-based rules

The methodology of extraction of corpus-based rules is described here. Automated analysis and classification of word groups was performed with the help of the Corpus of the Contemporary Lithuanian Language and morphological analyzer *Lemuoklis* (Zinkevičius 2000). Automatic analysis of word groups consists of the following stages: detection of text units, lemmatization of isolated words, classification of text units into relevant word groups.

The output of the lemmatization is shown bellow:
<w l="eiti(eina,ėjo)" m="finite verb, infinitive" l="eiti(eina,ėjo)" m="finite verb, participle>**eiti**<w>
After the lemmatization we have a morphological output that is used as an attribute for further classification. The main criteria of the classification are the length of a word group and possible word relations, i.e. subject, predicate, object, attribute and modifier relations in a group. After the revision of the classified text units by an expert the final structures were defined.

5. Conclusions

Our attempts to create the rules for formal syntactic analysis concentrates on the proposed methodology for the extraction of syntactic rules for formal grammar of the Lithuanian language. Subject, predicative, object, attribute and modifier groups were described and analyzed within two levels of the analysis. We also defined the necessary parameters and additional features that would increase the quality of automatic syntactic analysis. Our future plans include DG application for the parser of Lithuanian language.

References

Grigonytė, Gintarė 2004. Dalinis sintaksinis lietuvių kalbos veiksmažodžių analizatorius (Partial Syntactic Analyzer of the Lithuanian Language). *Bachelor thesis*. Vytautas Magnus University, Kaunas.

Grigonytė, Gintarė, Rimkutė, Erika 2005. Automatinis lietuvių kalbos veiksmažodžių grupių atpažinimas (Automatic Verb Phrases Recognition in the Lithuanian Language). In: *Informacinės technologijos 2005*, Kaunas: Kauno Technologijos universitetas. 315–320.

Mel'čuk, Igor 1988, Dependency Syntax: Theory and Practice. Albany: State University of New York Press.

Rimkutė, Erika 2003. Morfologinio daugiareikšmiškumo tipologija (the Typology of Morphological Ambiguity). In: Merkys, V.; Ambrazas, V.; Sauka, L. (eds.) *Lituanistica* 4 (56). 60–78.

Zinkevičius, Vytautas 2000. *Lemuoklis* – morfologinei analizei (Morphological analysis with *Lemuoklis*). In: Gudaitis, L. (eds.) *Darbai ir Dienos* 24. 246–273.

GINTARĖ GRIGONYTĖ is an engineer-programmer of the Centre of Computational Linguistics at Vytautas Magnus University. She is a student of Software engineering programme at Kaunas University of Technology. Her Master thesis deals with a dependency analysis for Lithuanian language. Her research interests include computational linguistics, software engineering, automatic syntactic analysis. E-mail: g.grigonyte@hmf.vdu.lt

ERIKA RIMKUTĖ is a junior researcher of the Centre of Computational Linguistics at Vytautas Magnus University. She received her M. A. (Lithuanian language) at Vytautas Magnus University. Her research interests include corpus linguistics, computational linguistics, automatic morphological analysis and synthesis, morphological ambiguity and disambiguation and automatic syntactic analysis. Her doctoral study focuses on morphological ambiguity and disambiguation in the Lithuanian language. E-mail: e.rimkute@hmf.vdu.lt.

EVALUATION OF LATENT SEMANTIC VECTOR MODELS USING A SWEDISH WORD COMPREHENSION TEST

Leif Grönqvist
GSLT (the Swedish Graduate School of Language Technology)

Abstract

This paper presents an evaluation of latent semantic vector models trained with different corpora and parameter settings. The important result is how the performance of the model changes depending on various parameters when the evaluation is not only the usual information retrieval testbed, but a word comprehension test. The new evaluation set may also be of interest for other researchers since evaluation sets for Swedish vector models is not very common.

Keywords: Latent semantic indexing, evaluation, semantic vector models, word comprehension test, Högskoleprovet, LSI, SVD

1. Introduction

Latent Semantic Indexing (LSI) has been around for a while since it was introduced to the world of information retrieval in the late 80's. (Deerwester et al. 1990) A well functioning Latent Semantic Vector Model (LSVM) has been proven to improve recall but also precision for document retrieval. There are two straight forward ways to use an LSVM in document retrieval:

- Use the model to expand the query with relevant terms before the query is sent to the retrieval system (Qiu and Frei 1993; Sahlgren et al. 2002)
- Use the model to calculate a vector corresponding to the query, and return documents having vectors (obtained by the model) close to the query vector (Telcordia 2003)

1.1. The vector space

A semantic vector space may be calculated automatically using some different methods, i.e. Random Indexing (RI) (Sahlgren et al. 2002) and Singular Value Decomposition (SVD) (Berry et al. 1995). The semantic vector space is a projection of the original co-occurrence based vector space but still has hundreds of dimensions. The methodology to calculate similarities (or distances) in a projected vector space is often referred to as Latent Semantic Indexing (LSI). (Deerwester et al. 1990)

1.2. A major weakness with the current use of LSI

Multi Word Units (MWUs) and documents are treated as bag of words. Obviously this is not optimal, since the meaning of the n-word unit $w_1..w_n$ is not just the sum of the meanings of the words. Vector addition is transitive and commutative, which would result in the same meaning for the units $w_1w_2w_3$, $w_1w_3w_2$, and $w_3w_2w_1$, so this approach can never be perfect. Hulth (2004) shows that less than 14% of all manually selected keywords contain just one token, so this problem should not be neglected if we want better precision and recall in document retrieval. One goal of this paper is to propose and evaluate some alternative ways to calculate the vectors corresponding to MWUs.

1.3. Evaluation

Used in an information retrieval context, LSI helps a lot despite the MWU problem. In this paper we present a new evaluation method which, used in combination with an IR testbed, will show how well a model handles both single word keywords and MWUs. We think that the two evaluation methods, a document retrieval task and a word comprehension test, will complement each other in a very nice way.

2. The evaluation method

We are using *the Infomap NLP System: An Open-Source Package for Natural Language Processing*, developed by the Computational Semantics Lab at Stanford University's Center for the Study of Language and Information. It is an implementation of singular value decomposition (SVD) and interface software to calculate a word by word matrix from a corpus, running the SVD. The interface is easy to use and also seems to work very well. The evaluations has been performed using an AWK-script, building the corpora, calling the Infomap system, and calculating the results. Similar experiments has been done using LSI and SVD on "The Test Of English as a Foreign Language" (TOEFL) with a reported correctness of 64%. (Berry et al. 1995) Slightly better results have been reported using random indexing instead of SVD. (Karlgren and Sahlgren 2001)

3. Parameter settings

In this section we will describe the parameters and the selected possible values.

3.1. Corpus choice

We believe a lot in the old corpus linguist's Mantra "there's no data like more data" since LSI rely a lot on redundancy. Unfortunately we are not able to process large corpora at the moment, due to weaknesses in the software package. Another important thing to keep in mind is that our evaluation set contains a lot of very rare words in newspaper text. On the other hand, this is just an evaluation and not a goal in itself. We will now describe the corpora used. We will use the term "types" for distinct word tokens and "tokens" for running words.

- **Newspapers**: We have a collection of around one million newspaper articles, in total 500 million tokens. From this collection, parts of various size are created: 1, 5, 10, 50, 100, and 500 million tokens. The 500 million token file contains more than 3 million types.

- **Bring thesaurus**: A Swedish thesaurus from 1930. It is inspired by Roget's thesaurus, and just like Roget, it has around 1000 main categories containing a main word and groups of related nouns, verbs, and adjectives. We have not used these structures at all so each category is counted as one document. In total, Bring contains 60 000 types and 140 000 tokens.

- **Lexin**: A dictionary for language learners of Swedish. It contains 19 000 main words with short descriptions. Each main word and its description is counted as a document. In total 200 000 tokens.

- **The Bible**: This is just a machine readable version of the Swedish Bible. It contains 800 000 tokens and 50 000 types, divided into 1200 documents – one for each chapter.

- **Swedish Parole**: A result of a project financed by the European Union. This Swedish part contains 20 million tokens and 600 000 types.

We have been using the corpora one by one for training but also combined them.

3.2. Input matrix

The Infomap software does not use a compact sparse matrix format, so the matrix has to fit into the RAM memory which in our case was 4 Gigabytes. In each cell there is a floating point number using 4 bytes which gives us a maximum matrix size of around one billion cells. Our choice of settings are: 200 000 x 5 000, 100 000 x 10 000, 100 000 x 5 000, 100 000 x 1 000, 50 000 x 20 000, and 50 000 x 10 000.

3.3. Context size

The context size decides how close two words have to occur to be seen as a co-occurrence. If the context size is large, all words in the current document will be seen as context words to each other. We have chosen to look at the values: 500, 100, 50, 30, 10, and 5 words.

3.4. Number of dimensions after projection

This is the number of dimensions in the new vector space obtained by the SVD. The number of dimensions is much lower than the vectors in the original co-occurrence matrix, but it is easy to calculate the vector in this space for any known word or combination of words, using the information from the SVD. It is reasonable to think that a bigger training corpus would need a higher dimensionality for all topics to fit into the vector space and earlier articles about LSI proposes a dimensionality from 50 to 500. We have chosen to look at: 50, 100, 200, 400 and 800.

3.5. Combining the parameters

A new computer performance problem arises when combining all the different parameter settings. Apart from the different training corpora, the different settings result in 180 combinations, and with some compositions of training corpora, the number of LSVMs to train easily reaches thousands. Each training is rather computation intensive since it contains calculations on a huge matrix and also requires hundreds of megabytes of disk space to store the model. We did not have the computer power for all these calculations, but exploring the parameter space works quite well even without all data points.

4. New ways to calculate the sum of meanings

A better way to look up MWUs in a semantic vector space would include a better way to calculate the true meaning of an MWU but just adding the meaning of each word it contains. Unfortunately the vector spaces we get from standard SVD or RI contains just single words if we have not made a better tokenization of the training data before calculating the vector space. Therefore, without preparing the data in a different way before running SVD or RI, the only obvious way to get a vector from an MWU is the sum of the vectors corresponding to each word.

4.1. Preparation of data before SVD

The baseline tokenization of the training data is to just create one token for each word. Delimiters between words are spaces and other non-letters. This process is fast and since this work is a pilot study for future work with large corpora containing billions of tokens, even the more complicated tokenization models has to be fast. We are aiming on a system architecture with a total processing time in the magnitude of 24 hours on a decent workstation.

4.1.1. The baseline tokenizer

Even the baseline tokenizer (BT) contains some difficulties. We need to recognize acronyms which may include '.' inside tokens. Other non-trivial tokens are dates, intervals, number (including spaces and/or a decimal point), etc. But, the baseline tokenizer will not try to find things like proper names, compounds [1] or fixed phrases.

4.1.2. The tuple extended tokenizer

This tokenizer (TUP) will take one more step from BT as an attempt to handle MWUs. The resulting string of BT-tokens will go through a tuple expander, adding all n-grams up to a specific length. The result will contain separate vectors for all these new tokens consisting of up to four words each. There is a risk of overtraining since the same BT-token will be a part of many TUP-tokens, but every original word is a part of the same number of tuples so we do not think this will be a problem. Compared to a noun phrase chunker or parser, TUP has the strength that it is language independent and very fast.

An alternative to TUP is to transform the training corpus to a set of n-character strings with a fix length. This will give the vector model a fair chance to make a guess on unknown words based on substrings instead of just random.

5. Presentation of the evaluation set (HP440)

The evaluation set is based on a Swedish word comprehension test (called ORD) from "Högskoleprovet" (an entrance test for university studies). There is a new test twice a year and our collection contains the exercises from 11 tests from the years 1998-2004, in total 440 queries with five alternative terms each.

5.1. Phrases in the queries

This test is not just a synonym test containing single words. Some of the query terms, and a bigger proportion of the alternative terms contain phrases. The query terms contain in average 1.17 tokens with a maximum of four tokens, but only 10.9% of the query terms contain more than one token. For the alternative terms the average number of tokens is 1.61, maximum is 10, and 35.5% contain more than one token.

[1] Since we primarily work with Swedish, compounds are not such a big problem. They are written without spaces between the parts, like in German.

5.2. About the words

One should note that ORD in “Högskoleprovet” is rather difficult. The average result was 62% among people who are trying to qualify for university studies. Many of the words are rare and/or old fashioned, and some are multi-word idioms.

6. Evaluation results

Since it has been practically impossible to calculate all combinations of LSVMs we had to explore the parameter space with partial information. On the other hand we had the possibility to calculate any specific LSVM we wanted. Here we will go through the parameter space, dimension by dimension, but still keep the other dimensions in mind since they may interfere with each other.

6.1. Corpus choice

We first tried each corpus one by one. The newspaper corpora of sizes bigger than 5 million got extremely bad result since only a small part of their vocabulary fitted into the input matrix. The Bible and Parole were not possible to use at all, by unknown reasons. The best results seem to come from LSVMs trained with Bring and Lexin put together. Bring alone gives a little bit lower results, followed by Lexin alone and then Bring, Lexin, and newspaper texts combined. The worse results came from the newspaper texts alone.

6.2. The input matrix size

The results of this evaluation shows that vocabulary is more important than the number of co-occurrence types, but this parameter is not very interesting since better LSI software would let us use the full matrix.

6.3. Number of dimensions

In earlier papers, the performance is said to increase up to a choice of around 300 dimensions, (Dumais 1995) and then slowly decrease for a dimensionality above the optimum. Our evaluation shows that the performance increases up to 400 dimensions, but the decrease is not significant when continuing to 800 dimensions.

6.4. Context size

The tested context sizes stretches from 5 to 500. It is difficult to guess which one is the best and it will probably vary a lot depending on the training corpus. The result in our tests shows that the context size 100 gives the best result, but the extremes 5 and 500 is not much worse, however the difference is statistically significant. This may change for other training corpus compositions.

7. Conclusion

We should say at once that some of the results were a bit disappointing since tuple expanding tokenizations did not improve the results at all.

7.1. The best results

Bring and Lexin in combination give the best results. With a context size of 100, rather high dimensionality of 400 or 800, and the biggest possible matrix size 100 000 x 10 000, which in this case fitted all types, we got a result of 58.8% correct answers which is far above the baseline 20%. For the known query terms, the result goes up to 65.1% on the remaining 367 queries. An attempt to add one million tokens of newspaper texts to the successful Bring+Lexin corpus degrades the results. Probably this is because all proper names and nouns replace the words important for the HP440 evaluation set.

7.2. Handling MWUs

One condition to make it possible to go on with this research is to have working, effective software. The attempts with tuples and character tuples were not very successful but it may work better with a bigger training corpus. Another strategy we want to try is to use an extremely fast dependency parser. (Nivre 2003) When we have a better LSI package and the various MWU handling methods are developed, we will try to optimize the settings for a high vocabulary LSVM useful in search engines. The evaluation used will be both the HP440 set and a document retrieval set with documents, queries, and relevance judgements, developed in Borås, Sweden. (Ahlgren 2004)

References

Ahlgren, Per 2004. *Ph.D. thesis*: University College of Borås and Göteborg University

Berry, Michael W.; Dumais, Susan T.; Letsche, Todd A. 1995. Computational methods for intelligent information access

Deerwester, Scott C.; Dumais, Susan T.; Landauer, Thomas K.; Furnas, George W.; Harshman, Richard A. 1990. Indexing by latent semantic analysis. In: *Journal of the American Society of Information Science* **41(6)**, 391–407

Dumais, Susan T. 1995. Using lsi for information filtering: Trec-3 experiments. In: *The Third Text REtrieval Conference (TREC3)*, National Institute of Standards and Technology

Hulth, Annete 2004. *Ph.D. thesis*: Stockholm University

Karlgren, Jussi; Sahlgren, Magnus 2001. From words to understanding. In: *Foundations of Real-world Intelligence* 294–308

Nivre, Joakim 2003. An efficient algorithm for projective dependency parsing. In: *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 03)*. 149–160

Qiu, Yonggang; Frei, Hans-Peter 1993. Concept-based query expansion. In: *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, Pittsburgh, US. 160–169

Sahlgren, M.; Hansen, P.; Karlgren, J. 2002. Sics at clef 2002: Automatic query expansion using random indexing. In: *Advances in Cross-Language Information Retrieval, Third Workshop of the Cross-Language Evaluation Forum, CLEF 2002*, Rome

Telcordia 2003. Telcordia TM Latent Semantic Indexing Software (LSI): Beyond Keyword Retrieval. Technical report: Telcordia Techologies

LEIF GRÖNQVIST has a M.Sc. in computing science, and is now a Ph.D. student in language technology at Växjö University in Sweden, writing a thesis on latent semantic indexing for information retrieval. Earlier experience includes work in Gothenburg on spoken language corpora for several years, and also cooperation with departments in Santiago de Cuba, Pretoria, and Copenhagen.

INFORMATION-SHARING AND CORRECTION IN ESTONIAN INFORMATION DIALOGUES: CORPUS ANALYSIS

Tiit Hennoste, Olga Gerassimenko, Riina Kasterpalu, Mare Koit, Andriela Rääbis, Krista Strandson, Maret Valdisoo

University of Tartu (Estonia)

Abstract

We are studying Estonian spoken dialogues with the goal to develop a dialogue system (DS). Two kinds of subdialogues have been analysed in information dialogues: information-sharing and correction. We analyse reasons and conditions of initialization of the subdialogues, their structure and linguistic features. The DS which is playing the role of information officer must be able to recognise a subdialogue initiation by client and to initiate and carry out subdialogues itself. A goal stack is used by the DS for dialogue processing.

Keywords: spoken dialogue, information-sharing, correction, dialogue system, goal stack.

1. Introduction

When a client is calling an information center and asking a question then the information provider not always is able to give an answer immediately. (S)he needs additional information to determine the client's goal precisely, and initiates an *information-sharing* subdialogue. Similarly, a client may start a *clarification* subdialogue if the answer does not satisfy his/her goal. Both of the partners can initiate *correction* subdialogues during a dialogue or a subdialogue.

These three kinds of subdialogues are differently understood by different researchers. Information-sharing is transfer of knowledge from one participant to another. Sometimes this kind of subdialogues is called knowledge precondition subdialogues because they are initiated by the agent to satisfy preconditions of a higher-level goal (Jurafsky, Martin 2000: 748). In this case, an agent tries to elicite knowledge from the partner (e.g. a travel agent asks details of a trip from a client). On the other hand, an information-sharing subdialogue can be initiated by an agent to evaluate a proposal of the partner (Chu-Carrol, Carberry 1995), e.g. a dialogue system is transferring its own knowledge to the user to resolve its uncertainty regarding the acceptance of a user proposal. In later publications negotiation is called a correction subdialogue (Chu-Carrol, Carberry 1998; Jurafsky, Martin 2000: 748). Correction is considered as a plan change (e.g. a client rejects a previous plan to travel on Friday and orders a ticket for Sunday), or error correction (Kirchhoff 2001). Clarification is considered as specification of answer (e.g. after the gate number is got from the information provider a client additionally asks for precise location of the gate), or as solving of communication problems (McTear 2004). Table 1 illustrates the different kinds of subdialogues (A, B – dialogue participants).

Table 1. Subdialogues of a dialogue

A: question/request	A: question/request	A: proposal
B: information-sharing	B: answer	*B: correction*
A:	*A: clarification/*	*(=negotiation)*
B: answer/grant	*(error) correction*	*A:*
	B:	B: accept/reject

In this paper, two kinds of subdialogues are considered: information-sharing and correction. Correction is understood as solving of communication problems, which is called repair in conversation analysis (CA). The difference between the two kinds of subdialogues is that information-sharing is ”looking forward”, i.e. sets up a subgoal of the primary goal (to get certain information) and therefore advances (works for progress of) a theme, while correction, or repair, is ”looking backward”, i.e. solves a problem (e.g. non-understanding) in previous text.

Our study is based on the Estonian dialogue corpus EDiC (Gerassimenko et al. 2004). The corpus contains about 600 human-human spoken dialogues, among them 328 information dialogues. Dialogue acts are annotated in EDiC using a typology which departs from the point of view of CA. The acts are divided into two big groups – adjacency pair (AP) acts (e.g. question–answer) and single (non-AP) acts (e.g. continuer). For this paper, 177 information dialogues, annotated in 2004, have been selected from EDiC. 127 information-sharing subdialogues were found in 64 dialogues. Correction subdialogues (88 in total) have been analysed in the same dialogues.

2. Information-sharing

Information-sharing is initiated always by the information provider (P) after a client’s (C) request (70% of cases) or question (30%). The purpose is to get additional information which is needed for answering. A subdialogue consists of one or more APs. It is typical for Estonian information dialogues that such a subdialogue consists of one question (offering answer or alternative question in most cases) followed by answer or (more rarely) of one directive (offer in our case) followed by agreeing. Example 1 illustrates an information-sharing subdialogue[1].

Example 1. Information sharing in a dialogue (marked with -->)

```
C: palun kodumasinate telefoninumber Pärnus.  | DIF: REQUEST |
number of home machines in Pärnu please
(1.2)
-->P: `kauplus Kodumasinad.   | QUF: OFFERING ANSWER |
the shop Home machines
-->C: jah.      | QUS: YES |
yes
```

[1] Transcription of CA is used in examples. Names of dialogue acts consist of two parts: the first two letters give abbreviation of the name of act-group, e.g. DI directives; the third letter is used only for AP acts - the first (F) or second (S) part of an AP act; 2) full name of the act (Gerassimenko et al. 2004).

```
(7.0)
-->P: `Talina maante `üks.  | QUF: OFFERING ANSWER |
Tallinn road one
-->C: jah.   | QUS: YES |
yes
P: `number > kinnitamata < `andmetel `neli `neli kolm?      | DIS: GIVING
INFORMATION |
the number is four four three
```

Adjusting conditions of answer P is obtaining details for information retrieval (e.g., if C asks for a bus station phone number then the town of his location must be specified), or is offering choices to C (e.g. general or business information of the requested institution), or (s)he makes a choice himself/herself and asks an agreement of C). Usually, information-sharing will specify an institution (name, location, structural unit) or is expecting choice/approval of a phone number. If the asked information is missing then P sometimes offers substituting information (e.g. a number of a secretary instead of the asked number of book-keeping). Information-sharing typically begins with a question offering answer (38%), another type of question (alternative, yes/no etc., 39%), or offer (23%). Full sentences and parts of sentences (phrases) are used almost equally (51% and 49%).

The first part of an AP used by P starting a subdialogue determines possible second parts which can be used by C. In our dialogues, C's agreement/yes mostly follows to P's offer/question (82%). This means that P correctly recognised the C's goal. If P asks an alternative question then C is able to choice an alternative in 77% of cases. Table 2 gives the most frequent structure of an information-sharing subdialogue. The acts given in parentheses can be missed.

Table 2. Location and structure of an information-sharing subdialogue

MAIN LINE
C: request/wh-question
(P: response, postponing the answer)
INFORMATION-SHARING
--> P: question offering answer/offer
--> C: yes/agreement
BACK TO MAIN LINE
(C: question)
P: giving/missing information

3. Correction

We differentiate three types of correction (repair) initiations. The first two types are *clarification* and *non-understanding:* the hearer initiates a correction and the partner carries it out. Both of the initiations indicate hearer's perceiving problem, non-understanding expects partner to repeat problematic part of his/her turn and clarification repeats the problematic part, expecting partner either to confirm or to correct this repetition. The third type is *reformulation*, where the hearer initiates a correction and suggests her own interpretation of the problematic place. The partner agrees with or rejects this interpretation. Therefore the hearer is not correcting a mistake here but indicates an understanding problem.

Correction subdialogues are initiated in certain limited cases, e.g. regarding information that must be exact (prices, concessions, e-mail addresses, actions that will be carried out next). The problems that cause correction principally can be located in an arbitrary past turn. In our subcorpus, corrections are initiated regarding the immediately previous turn in 90% of cases (Example 2).

Example 2. Correction in a dialogue (marked with -->)

```
C: (.) `tahtsin küsida `Tartus=e (.) `Kalda tie `kolmkümmend (.) `pleki
`ukse `koda (.) kas te `saate mu anda.    | QUF: OPEN Y/N |
I would like to ask Tartu Kalda street thirty tin door shop could you
give me
(1.2) on sellised (.) {teil või} £      | QUF: Y/N |
do you have such
(4.2)
-->P: ja `aadress oli `Kalda `tee? | QUF: WH | | RRF: NON-UNDERSTANDING |
and address was Kalda street
(.)
-->C: £ `kolmkend kui ma `õieti `mäletan. £   | QUS: GIVING INFORMATION |
| RRS: PERFORMING |
thirty if I remember correctly
(14.5)
P: `Kalda tee kolmkümmend on: (.) `Pee `Haa Pro`jekt,    | QUS: GIVING
INFORMATION |
Pee Haa Project is in Kalda street thirty
(.)
```

C initiates corrections a little more than P (57% and 43%, respectively). There are differences in different correction initiations (Table 3).

Table 3. Correction initiations

Correction initiation	**Client**	**Provider**
clarification	73%	27%
non-understanding	70%	30%
reformulation	28%	72%

Clarification forms 61%, non-understanding 23%, and reformulation 16% of correction initiations. Most of the clarifications used by C are repeats of phone numbers. It is typical in calls that clients repeat a phone number to be sure that it has been understood correctly.

There are two typical locations of a correction subdialogue: after C's question/request and P's answer. They give us two prototypical structures (Table 4).

Table 4. Location and structure of a correction subdialogue

A. Problem in question/request	**B. Problem in answer**
MAIN LINE: C's question/request CORRECTION --> Correction initiation by P --> Problem solving by C BACK TO MAIN LINE: P's answer/grant	MAIN LINE: P's answer CORRECTION --> Correction initiation by C --> Problem solving by P BACK TO MAIN LINE: C's response (*mhmh/ ahah /jah*) + (new) question/request/finishing conversation

70% of corrections that concern C's question/request belong to the prototype A. 80% of corrections regarding P's answer belong to the prototype B. In human-computer interaction, there will be more communication problems than in human-human conversation, and the DS will initiate more corrections.

4. Modelling subdialogues

A stack is an appropriate data structure to describe setting up and abandoning of goals, shared between client and DS (information provider). C's first question/request sets up the main goal which is put on the bottom of the stack. The following information-sharing questions set up new goals which are going into the stack step by step. To achieve the main goal, all the goals in the stack that are located higher itself must be achieved and removed. If the stack is empty then all the goals are achieved.

Starting a correction after C's request, DS puts a goal into the stack only after the correction is performed. Similarly, if C starts a correction subdialogue after getting an answer, then the goal is remained in the stack until the communication problem is solved. This is why it seems useful to distinguish "backward-looking" correction and "forward-looking" information-sharing.

5. Conclusion and Future Work

Information-sharing is transfer of knowledge from one partner to another to achieve a common goal in a co-operative dialogue. A typical subdialogue consists of one adjacency pair of dialogue acts (question offering answer/offer – yes/agreeing). A subgoal of the main goal is set up and achieved during a subdialogue. Correction can be used for solving communication problems by both of participants regarding both a question or answer. A typical subdialogue initiated by client is clarification of a phone number.

Our future work concerns implementation of the stack structure in a DS.

Acknowledgement

This work is supported by the Estonian Science Foundation (grant 5685).

References

Chu-Carrol, Jennifer; Carberry, Sandra. 1998. Collaborative response generation in planning dialogues. In: *Computational Linguistics*, 24(3). 355–400.

Chu-Carrol, Jennifer; Carberry, Sandra. 1995. Generating information-sharing subdialogues in expert-user consultation. In: Proc. of *IJCAI.* Retrieved February 13, 2005, from http://arxiv.org/abs/cmp-lg/9701003.

Gerassimenko, Olga; Hennoste, Tiit; Koit, Mare; Rääbis, Andriela; Strandson, Krista; Valdisoo, Maret; Vutt, Evely. 2004. Annotated dialogue corpus as a language resource: an experience of building the Estonian dialogue corpus. In: *The first Baltic conference "Human language technologies. The Baltic perspective". Commission of the official language at the chancellery of the president of Latvia*. Riga. 150 – 155.

Jurafsky, Daniel; Martin, James H. 2000. An introduction to natural language processing, computational linguistics, and speech recognition. Prentice Hall.

Kirchhoff, Katrin. 2001. A comparison of classification techniques for the automatic detection of error corrections in human-computer dialogues. In: *Proceedings of the NAACL Workshop on Adaptation in Dialogue Systems* , Pittsburgh, PA. Retrieved February 13, 2005, from http://ssli.ee.washington.edu/people/katrin/-Papers/naacl01.pdf

McTear, Michael F. 2004. Spoken dialogue technology: toward the conversational user interface. Springer Verlag, London.

TIIT HENNOSTE is lecturer of Estonian literature and culture at the department of Finno-Ugric Studies, University of Helsinki, Finland, and researcher of the department of Estonian and Finno-Ugric Linguistics, University of Tartu, Estonia. He received his M.A. (journalism) at the University of Tartu. His research interests concern spoken Estonian, CA, and corpus linguistics. His doctoral study focuses on grammar of spoken Estonian. He is the member of the editorial board of the Estonian journals Keel ja Kirjandus, Akadeemia, Oma Keel, and the internet-journal Journal of Intercultural Communication (Gothenborg). E-mail: tiit.hennoste@helsinki.fi.

OLGA GERASSIMENKO is M.A. student of the University of Tartu. She received her B.A. (Estonian language) at the same university, dealing with CA of spoken Estonian and Russian. E-mail: gerro@ut.ee.

RIINA KASTERPALU is Ph.D. student of the University of Tartu. She received her B.A. (general linguistics) at the same university, dealing with structure of spoken dialogue. Her research interests concern spoken language, especially dialogues. E-mail: riina.kasterpalu@ut.ee.

MARE KOIT is professor of the Institute of Computer Science, University of Tartu. She received her candidate of sciences degree (mathematics) at the Academy of Sciences of the USSR (Moscow), dealing with mathematical models of semantics and text generation. Her research interests concern human-computer interaction. E-mail: mare.koit@ut.ee.

ANDRIELA RÄÄBIS is researcher and Ph.D. student of the department of Estonian and Finno-Ugric Linguistics, University of Tartu. She received her M.A. (Estonian language) at the same university, dealing with CA of spoken Estonian. E-mail: andriela.raabis@ut.ee.

KRISTA STRANDSON is researcher of the Institute of Computer Science, University of Tartu. She received her M.A. (Estonian language) at the same university, dealing with CA of spoken Estonian. Her research interests concern spoken language, especially text planning and school interaction. E-mail: krista.strandson@ut.ee.

MARET VALDISOO is Ph.D. student of the University of Tartu. She received her M.Sc. (computer science) at the same university, dealing with dialogue modelling. E-mail: maret@ut.ee.

VOCABULARY TRAINING PROGRAM USING TTS AND SPEECH RECOGNITION TECHNOLOGIES

Michael Hofmann[*], **Boris Lobanov**[†]
[*]University of Technology, Dresden, Germany
[†]United Institute of Informatics Problems, Minsk, Belarus

Abstract

Learning the right pronunciation is one of the most difficult tasks in mastering a foreign language. Common (commercial) PC based training applications provide only limited support for accentuation and listening or speaking. The aim of this paper is to present a vocabulary learning application that uses freely available speech synthesis and recognition technology. Various programs are analyzed whether and how they can be integrated in such a system. Different modes for the evaluation of recorded speech are presented and analyzed in their ability to judge pronunciation and accentuation correctness. It is shown that while synthesis support is readily available, the used approach of recognition-by-synthesis imposes severe limits to the ability of the system to generate meaningful pronunciation rates.

Keywords: text to speech, memory, synthesis, pronunciation, assessment

1. Introduction

1.1. Language learning

The knowledge of a foreign language consists of several parts that can't be learned independently (Neri et al. 2002): grammar, pronunciation, vocabulary and others. Whereas the first has to be learned interactively lead by a tutor or teacher to be efficient, e.g. by taking part in language training courses offered by a university or language school, vocabularies and their pronunciation can in part be learned autonomously.

Language training therefore should cover all this parts through different means of practice material and support: dialogs simulating every-day situations that appear in real life, the interactive construction of sentences to practice grammar, Listen and repeat exercises to train understanding and pronunciation, vocabulary drills for memorizing.

1.2. Task definition

To assist the learning process of a foreign language, a vocabulary training program with the support of freely available speech synthesis and recognition systems is developed.

Assistance is given in all of the following steps in learning words and phrases in a foreign language: listen to them, be able to understand them, to write them down, to translate them into the native language and to speak them with correct pronunciation.

To achieve this, synthesized words and phrases have to be recognized and translated by the student. The right accentuation is obtained from a database and presented to the student while learning. Afterwards the student is requested to speak the phrase, which is recorded and evaluated regarding pronunciation. Training sessions can be easily created and multiple training modes are supported. Support for German, English and Russian synthesis is available. The program can be run on current Unix and Windows operating systems.

Special attention is paid to the use of freely and widely available components. To achieve reasonable portability and good performance, the program is written in C++ using common platform-independent class libraries and toolkits. To implement the Graphical User Interface (GUI), the GTK+-Toolkit was used.

2. Learning process

The here presented training system focuses on a single task: to enable a student of a foreign language to learn and memorize vocabularies and their pronunciation effectively.

Among the different human memory systems the long term memory is responsible for the storage of declarative knowledge like vocabularies. Because the content is subject to the natural forgetting process, repeated recalls are necessary for information to last for longer.

Repetitio mater memoriae—Repetition is the mother of memory. Therefore the most important part of vocabulary training is the repetition of them in different ways until it can be assumed that the student memorized them completely.

With the increasing consolidation of the trained vocabularies, the time intervals in which material is reviewed can be increased over time using a technique called *spaced repetition*.

3. Program structure

3.1. Features

The program features an easy to use interface to be able to enter new training data and to train a number of vocabularies repeatedly. Acoustic versions of the entered vocabularies and stress positions are automatically provided if support for this feature in the target language is found. The repetition rate and training mode can be adjusted to the students needs. The knowledge level for each word or phrase is saved between sessions.

Experimental support for feedback of pronunciation quality and the level of correctness is available.

3.2. Synthesis support

In contrast to commonly available language learning programs, the auditory input to the student is not retrieved from a store of prerecorded examples. To allow the use for different languages and environments, speech synthesis is used for the generation of an acoustical and phonemic representation of the training material for the current exercise.

Widely available speech synthesis packages include Mbrola (Dutoit et al. 1996) and Festival (Black and Taylor 1997), which provide support for many different languages.

The Festival package is provided under the revised BSD license. It provides support for English and Spanish voices and offers all functions of a TTS system.

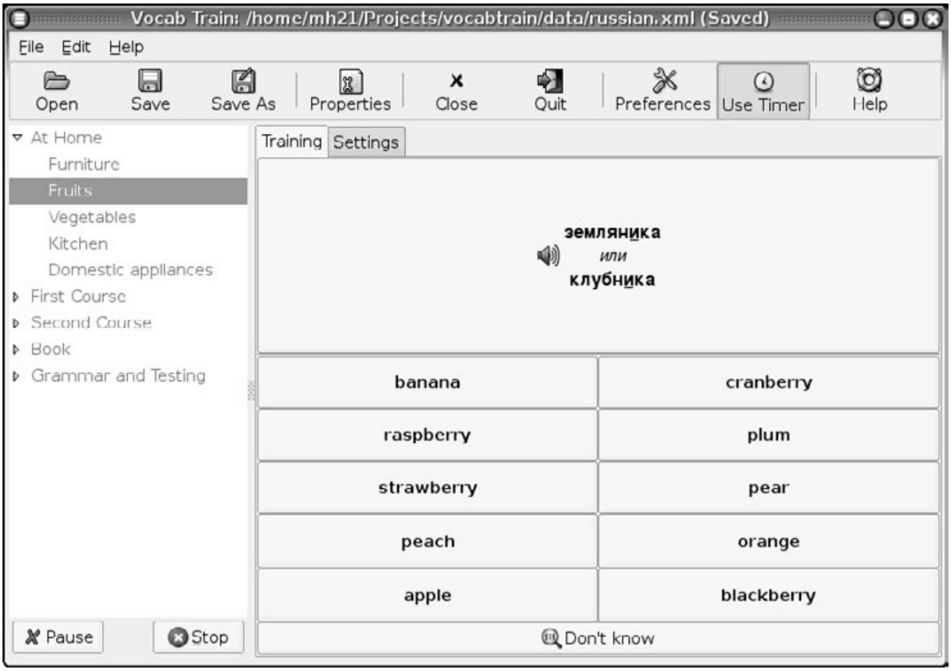


Figure 1: The training program in action

The Mbrola program is distributed under non-free license terms allowing non-comercial and non-military use only. It provides only the phone to wave conversion and can be combined with different text to phone converters to create a full text to speech system. It can be used with a wide range of languages.

For support of German and Russian a common TTS-synthesis prototype is used (Lobanov et al. 1998).

The attainable speech quality from nowadays available speech synthesizers is far good enough to be used in training programs. Although also longer training material is imaginable, the problems of current speech synthesis technology with more or less monotonic speech patterns because of inadequate prosodic control do not affect the training on word and phrase level.

4. Learning modes

The program provides several possibilities to train vocabulary knowledge. Both the input mode (how the current exercise is presented to the student) and the answer mode (the way in which the student can supply the answer) can be selected (table 1).

The input for the student can be provided in several ways:

- Written. The question is printed to the screen. Additionally trains the reading for languages in different alphabets, e.g. Russian or Greek.
- Acoustic. The vocabulary to be translated is converted to auditory output and played. The playback can be repeated if desired. Will improve the students ability to recognize trained words and phrases in conversation.

Table 1: Possible training modes

Question	Answer	Training effects
Acoustic	Select other language	Understanding, translation
Acoustic	Select same language	Understanding, letter-phoneme relation
Acoustic	Write other language	Understanding, translation, spelling
Acoustic	Write same language	Understanding, letter-phonome relation, spelling
Written	Select other language	Reading, translation
Written	Write other language	Reading, translation, spelling

- Combined. This gives additional training effects in languages like French or English, where no direct letter-phoneme relation exists.

The student can choose between three different modes to answer the question:

- Select among a certain number of alternatives. In this way, a large number of vocabularies can be reviewed in a short time.
- Provide the answer by entering it directly in the same language as the question. Combined with the speech synthesis, it is possible to train only the understanding or disambiguation of words.
- Same as before, only use the other language.

5. Recognition and Scoring

To evaluate the pronunciation of the exercises by the student, the following four steps must be performed (Ambra Neri et al. 2003):

- Speech recognition: The incoming signal has to be recorded, processed and recognized.
- Scoring: Pronunciation evaluation based on different properties of the recognized speech.
- Error detection: The program isolates certain phones or parts of the utterance that do not match the stored representation with a certain confidence.
- Error diagnosis: The type of error made is identified.

The information obtained in the last three steps is then presented to the student. Care has to be taken that the student is actually able to understand the displayed information to correct his pronunciation accordingly.

5.1. Speech recognition

Although there are a couple of speech synthesis packages with a large number of different voices available, very few speech recognition systems can be obtained freely. Worse, most of these systems have to be trained to a certain speaker and are very limited in the number of languages with which they can be used.

To minimize the dependencies on speech technology software used by the system, the method of recognition-by-synthesis is used to prepare the speech signal patterns to be recognized. This only requires the same speech synthesis package as used beforehand.

For the recognition to work independently from the used audio equipment, the characteristics of the channel: student – microphone – sound card, has to be corrected to match the characteristics of the synthesis. The assumption is made that that characteristics of the given channel are time-invariant.

The spectrum of a longer utterance by the student in his native language is averaged over the whole time and compared to a similar obtained reference from the speech synthesizer. Each of these averaged spectra is regarded as the frequency response of the corresponding channel. The difference between them in the log domain is used for a band filter to calculate the attenuation for each separate band (Young 1996).

To align the synthesized reference and the student utterance, silence at the beginning and the end of each signal is stripped using a threshold of -40 dB.

Both signals are then converted to a sequence of feature vectors. Implemented algorithms are a Fourier transformation, a mel spaced filterbank and MFCC coefficients. Dynamic time warping is used to fit the students utterance to the reference signal.

5.2. Error scoring and feedback

A lot of work is done in the field of automatic pronunciation assessment (Cucchiarini et al. 1998; Neumeyer et al. 2000; Teixeira et al. 2000). The term pronunciation covers a wide range of speech properties from segmentation to word stress. Erroneous pronunciation can mean a deviation in any of the following (Cucchiarini et al. 1997): the fluency of the utterance, the used syllable structure, word stress, kind of intonation and the segmental quality.

Not all of them can be directly measured. Some observable factors are the overall speech rate, the phonetic segmentation, the phone selection by the speaker and the pitch contour.

In the context of the training program and the used recognition method, possibilities are limited: the overall speech rate and the pitch contour are meaningless because of the only use of words and short phrases. Phone selection can only be judged from the similarity of reference and student signal, not by comparison with other similar sounding, but wrong phones, which makes scoring more difficult.

To evaluate the deviation of the segment duration from the reference, the following equation is used. L_{user} and L_{ref} denote the total length of the user and reference signal and l_i the length of the corresponding phone segments.

$$S = \sum_{i}^{N} \left| \ln \left(\frac{l_{i,user}/L_{user}}{l_{i,ref}/L_{ref}} \right) \right|$$

Neri et al. (2002) examines different available Computer Assisted Pronunciation Training (CAPT) courseware on the way they provide feedback for pronunciation errors. It is concluded that feedback should be limited to a simple grade of correctness and a highlighting of the incorrect areas in the utterance.

5.3. Summary and problems

This paper presented a flexible configurable vocabulary training application that can be configured to work with various available synthesizing systems.

Although synthesis support is easy to implement and aids in learning a foreign language, the reliable evaluation of the students own pronunciation is very desirable.

The used approach of recognition-by-synthesis provides a possibility to judge the phonemic segmentation of utterances by the student in comparison to the synthesizer. Problems like the spectrum distortion by slow amplitude changes, tolerances of the found phone borders and difficulties in distinguishing successive consonants need to be solved to generate meaningful scores.

References

Ambra Neri; Catia Cucchiarini; Strik, Wilhelmus 2003. Automatic speech recognition for second language learning: How and why it actually works. In: *Proceedings of 15th International Congress of Phonetic Sciences*, Barcelona, Spain. 1157–1160

Black, A.; Taylor, P. 1997. The Festival speech synthesis system. University of Edinburgh

Cucchiarini, C.; de Wet, F.; Strik, H.; Boves, L. 1998. Assessment of dutch pronunciation by means of automatic speech recognition technology. In: *Proceedings ICSLP '98*, Sydney, Australia. 751–754

Cucchiarini, C.; H. Strik, H.; Boves, L. 1997. Automatic evaluation of Dutch pronunciation by using speech recognition technology.. In: *Proc. of IEEE ASRU*, Santa Barbara. 622–629

Dutoit, T.; Pagel, V.; Pierret, N.; Bataille, F.; van der Vreken, O. 1996. The MBROLA project: towards a set of high-quality speech synthesizers free of use for non-commercial purposes. In: *Proceedings ICSLP '96*: Vol. 3, Philadelphia, PA. 1393–1397

Lobanov, B.; Hoffmann R.; Ivanov, A.; Kubashin, A.; Levkovskaja, T.; Helbig, J.; Jokisch, O. 1998. A bilingual German / Russian text-to-speech system. In: *Proceedings of the 3nd International Workshop "Speech and Computer" SPECOM'98*, St. Petersburg. 327–330

Neri, A.; Cucchiarini, C.; Strik, H.; Boves, L. 2002. The pedagogy-technology interface in computer assisted pronunciation training. In: *Computer Assisted Language Learning* **15(5)**, 441–447

Neumeyer, L.; Franco, H.; Digalakis, V.; Weintraub, M. 2000. Automatic scoring of pronunciation quality. In: *Speech Communications* **30(2-3)**, 83–94

Teixeira, C.; Franco, H.; Shriberg, E.; Precoda, K.; Sonmez, K. 2000. Prosodic features for automatic textindependent evaluation of degree of nativeness for language learners

Young, Steve 1996. A review of large-vocabulary continuous speech recognition. In: *IEEE Signal Processing Magazine* **13(5)**, 45–57

MICHAEL HOFMANN is student of the University of Technology, Dresden, Germany. He is an invited researcher at the United Institute of Informatics Problems Nat. Ac. of Sc., Belarus.

BORIS LOBANOV is head of the Speech Recognition and Synthesis Laboratory at the United Institute of Informatics Problems Nat. Ac. of Sc., Belarus and Professor at the University of Bialystok, Poland. He has a Dr. Sc. degree in text-to-speech synthesis (1984). He is member of the European Speech Communication Association since 1995. His sphere of interests include TTS-synthesis, speech analysis and recognition and speech technology applications. He has written more then 200 publications in the area of speech synthesis and recognition.

ON MULTIMODAL ROUTE NAVIGATION IN PDAS

Topi Hurtig, Kristiina Jokinen
University of Helsinki, Finland

Abstract

One of the biggest obstacles in building versatile natural human-computer interaction systems is that the recognition of natural speech is still not sufficiently robust, especially in mobile situations where it's almost impossible to cancel out all irrelevant auditory information. In multimodal systems the possibility to disambiguate between several input and output modalities can substantially increase the intelligibility of dialogues and the robustness of interaction. The combination of natural speech and tactile gestures as input mediums, especially in map-based systems, have shown prominent results, although mature commercial applications are still to be developed. In this paper we present the MUMS Multimodal Route Navigation System, intended for public transportation commuting. The system allows users to present route queries with any preferred combination of speech and pen input, and the system provides navigational information via speech and graphical map representations. The focus of this document is on the system's natural interaction model, which is designed keeping in mind the current limitations in natural speech recognition.

Keywords: human-computer interaction, multimodal dialogue systems, natural language, modality fusion, user interfaces, mobile systems, route navigation

1. Introduction

Multimodal interactive systems have gained ground in recent years, and they do seem to provide a user-friendly alternative to several application fields. However, due especially to the lack of robustness in speech recognition, there is still a long way to go before any of these kind of applications pass as their human counterparts. The naturalness feature can be attached, not only to human-human communication, but also to applications that take advantage of users' natural ways of giving and receiving information. Natural interaction does not also only include verbal communication: much of the information content in human-human situations is conveyed by non-verbal signs, gestures, facial expressions, etc. Thus, in order to develop next generation human-computer interfaces, it is necessary to work on technologies that allow multimodal natural interaction: it is important to investigate coordination of natural input modes (speech, pen, touch, eye movement, etc.) as well as multimodal system output (speech, sound, graphics, etc.), ultimately aiming at intelligent interfaces that are aware of the context and user needs, and can utilize appropriate modalities to provide information tailored to a wide variety of users. Natural interaction could be considered an approach by which various users in different situations could exploit the strategies they have learnt in human-human communication.

2. Related research

Speech and tactile input are known to be very closely coupled, and their combined use has been extensively studied. For example in studies conducted with the QuickSet system (Oviatt et al. 2000; Oviatt 2001) it has been found that multimodal input can indeed help in disambiguating input signals, which improves the system's robustness and performance stability. Other advantages of multimodal interfaces include the ability to choose an input approach best suited for each person and each situation. Different modalities offer different benefits, and also the freedom of choice (Gibbon et al. 2000). Jokinen and Raike (2002) also point out that multimodal interfaces have obvious benefits for users with special needs who cannot use some or all the communication modes.

A clear disadvantage presented by multimodal interfaces is the special attention needed by the user in coordinating the input modalities, possibly resulting in cognitive overload. Also when receiving multimodal information, the user experiences stimulation of multiple senses, which also affects the cognitive load. From the system-centric view, multimodality requires advanced processes at the combination level and especially at the interpretation level, and also an adaptable approach to presenting information.

The system described in this paper is based on the USIX Interact project (Jokinen, et al. 2002) which aimed at studying methods and techniques for rich dialogue modelling and natural language interaction. In this follow-up project, the main goal of research is to integrate a PDA-based graphical point-and-click interface with the user's speech input, and to allow the system to output in speech as well as drawing on the map. Besides the technical challenges, an important goal is also to investigate possibilities for natural interaction in a route navigation task where the system is to give helpful information about the route and public transportation.

3. Multimodal interaction with MUMS

3.1. User interface

The system can perform two tasks: provide timetable information for public transportation and provide navigation instructions for the user to get from a departure place to a destination. The client application accepts speech and tactile input, and presents information via speech and graphical map data. The touch-screen map interprets all tactile input as locations, so a tap on the screen denotes a pinpoint coordinate location, whereas a circled area will be interpreted as a number of possible locations. The map can be freely scrolled and zoomed in real time, and the inputs are recorded simultaneously and time stamped for later modality fusion phase processing.

In order for the system to be able to retrieve route information, at least the departure and arrival locations must be known, which results in the system returning a route summary containing the most relevant route details. If the user does not provide all necessary information to execute a full route query, the system prompts the user for the missing information. As shown in Example 1 and Figure 1, the user can provide a segment of information either by voice or a map gesture. When all necessary information has been collected, the system will fetch the route details.

Example 1 (dialogue). The user presents a route query, makes a correction, and finally iterates departure times until a suitable route is found.

U: Uh, how do I get from the Railway station ... uh.
S: *Where was it you wanted to go?*
U: Well, there! + <map gesture>
S: *Tram 3B leaves Railway Station at 14:40, there is one change. Arrival time at Brahe Street 7 is 14:57.*
U: When does the next one go?
S: *Bus 23 leaves Railway Station at 14:43, there are no changes. Arrival time at Brahe Street 7 is 15:02.*

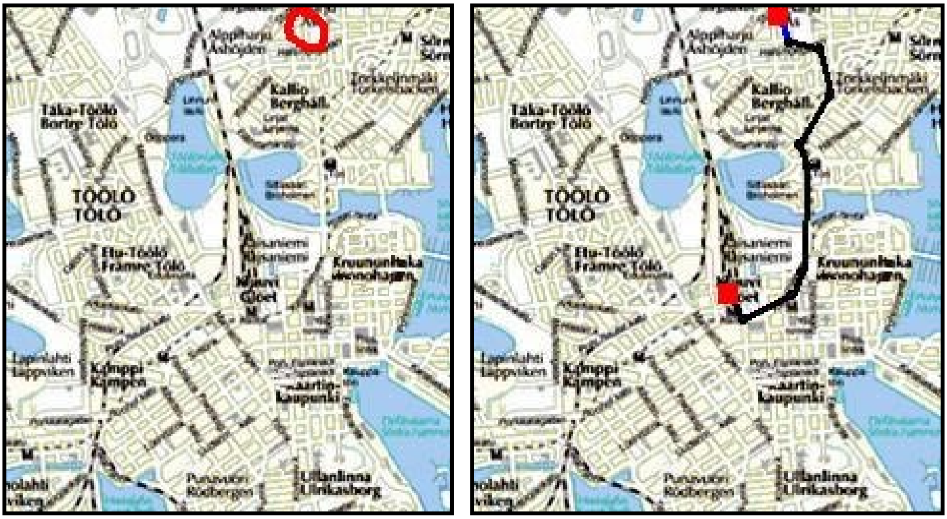


Figure 1. Tactile input (at left) and a graphical representation of a route (at right).

During navigation the route is presented on the screen (Figure 1, at right), and details are supplied by speech, as shown in Example 2.

Example 2 (dialogue). The user accepts the route suggestion and asks the system to instruct him/her on that route.

U: Ok. Navigate.
S: *Take bus 23 at the Railway Station at 14:43.*
U: Navigate more.
S: *Get off the bus at 14:49 at the Brahe Street stop.*
U: Navigate more.
S: *Walk 200 meters in the direction of the bus route. You are at Brahe Street 7.*
U: Okay.

In addition to requesting route guidance, users can also present questions about route details: travel times, distances, the stop count, etc. Users are also not restricted to any specific timing or form of input, and they can also make corrections to already submitted input at any dialogue phase.

3.2. Interaction model

Coherent frameworks for multimodal interaction patterns are yet to be formed, and thus application development is guided by a few selected studies (e.g. Oviatt et al. 2004), and also by work in the field of linguistics. The purpose of multimodal interfaces is to provide users with a more natural way to interact with a system. In practice, due to the deficiencies in the robustness of natural speech recognition, it is of utmost importance to design the interaction model so that the user is limited to a certain amount of possible ways of presenting information, but at the same time feels that he/she still is free to present this input in a natural and flexible way. The possibility to choose an input strategy is one of the most important factors accounting to naturalness during interaction (Oviatt 2001). The cognitive load experienced by a user is one way to measure an aspect of naturalness. Cognitive load increases e.g. in situations where the user must perform multiple simultaneous tasks or formulate complex utterances.

In MUMS, because of the rather limited functionality and task-specific nature of the system, the user is already limited to a handful of ways of forming a spoken route enquiry, which simplifies the recognition and interpretation processes. The first turn in the dialogue is initiated by the user, who is expected to present a route query. Even though there is just a handful of possible ways of formulating a query, this is clearly the critical point in the dialogue from the performance point of view. In case the user's utterance was misinterpreted or its contents insufficient, the system prompts the user for the missing or additional information one concept at a time in such a way that the user's cognitive load does not affect his/her output.

Several current multimodal applications trade naturalness for robustness by using explicit confirmations. A reliable way to confirm simple details, this approach is however, as Boyce (1999) points out, usually found inflexible and annoying. In the MUMS system, all confirmations are carried out in an implicit manner, as shown in Example 1. This approach is expected to be quite successful here, since the task at hand contains only a few variables. The route summary, presented by speech and graphics, is a straightforward and quick way to determine if the system interpreted his/her input correctly. The next user utterance could e.g. be: *"No, I wanted to get to the opera house"*.

Natural multimodal navigation is a field of research that has resulted in several practical applications. Map representations are a natural way of presenting spatial information, and speech can be used for guidance when the user's eyes are occupied with a mobile task. The success of a guidance task depends also on the naturalness and the cognitive load experienced by the user. The cognitive load can be reduced e.g. by presenting the information in suitable chunks (Cheng et al. 2004). Also important are the way the information is split into the output modalities, and the used level of detail.

In our approach graphical output is at the moment simple and static; a fetched route is kept on the map as long as the user is en route. Users can however choose the detail of the spoken instructions. The default detail level is intended for experienced commuters, and consists of just the basic details, e.g. times and locations. In the detailed level, information is presented in smaller chunks and the user is provided with additional details, such as the number of stops, travel times, etc. The detailed level is aimed at users needing clearer instructions, such as the visually impaired. At the moment the navigation level is set by the user, but we can also envisage that it would be possible for the system to adapt itself, by its knowledge of the particular situation and learning through interaction with the user, when to switch to a more detailed navigation mode.

4. Conclusion

The presented MUMS system provides the user with a natural way to enquire locational information and route navigation. The system's interaction level is designed for ease of use and naturalness, keeping in mind the challenges set by especially the current state of speech recognition. We assume that the system architecture is general enough to be used in other similar multimodal applications as well.

Further studies will aim at improving the integration and synchronisation of information in multimodal dialogues. The system will also be extended to handle more complex pen gestures, such as areas, lines and arrows. As the complexity of input increases, so does the task of disambiguation of gestures with speech, which will undoubtedly present us with new challenges. An early evaluation of the prototype system, including usability testing, will be conducted in the near future.

References

Boyce, S. 1999. Spoken natural language dialogue systems: User interface issues for the future. In: Gardner-Bonneau (ed.). *Human Factors and Voice Interactive Systems*. 37–62.

Cheng, H.; Cavedon, L.; Dale, R. 2004. Generating Navigation Information Based on the Driver's Route Knowledge. In: Gambäck B.; Jokinen, K. (eds.) *Procs of the DUMAS Final Workshop Robust and Adaptive Information Processing for Mobile Speech Interfaces, COLING-2004 Satellite Work-shop*, Geneva, Switzerland. 31–38.

Gibbon, D.; Mertins, I.; Moore, R. 2000 (eds.). Handbook of Multimodal and Spoken Dialogue Systems. Resources, Terminology, and Product Evaluation. Dordrecht: Kluwer.

Jokinen, Kristiina; Kerminen, A.; Kaipainen, M.; Jauhiainen, T.; Wilcock, G.; Turunen, M.; Hakulinen, J.; Kuusisto, J.; Lagus, K. 2002. Adaptice Dialogue Systems – Interaction with Interact. In: *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue.* Philadelphia, USA: Association for Computational Linguistics.

Jokinen, Kristiina; Raike, Antti 2003. Multimodality – technology, visions and demands for the future. In: *Proceedings of the 1st Nordic Symposium on Multimodal Interfaces*. Copenhagen.

Oviatt, Sharon; Cohen, P.R.; Wu, L.; Vergo, J.; Duncan, L.; Suhm, B.; Bers, J.; Holzman, T.; Winograd, T.; Landay, J.; Larson, J.; Ferro, D. 2000. Designing the User Interface for Multimodal Speech and Pen-based Gesture Applications: State-of-the-Art Systems and Future Research Directions. In: *Human Computer Interaction*, 15(4). 263–322.

Oviatt, Sharon 2001. Advances in Robust Processing of Multimodal Speech and Pen Systems. In: Yuen, P.C. and Yan, T.Y. (eds.). *Multimodal Interfaces for Human Machine Communication*. London, UK: World Scientific Publisher.

Oviatt, Sharon; Coulston, R.; Lunsford, R. 2004. When Do We Interact Multimodally? Cognitive Load and Multimodal Communication Patterns. In: *Proceedings of the Sixth International Conference on Multimodal Interfaces (ICMI 2004)*, Pennsylvania, USA. 14–15.

TOPI HURTIG is a researcher at the University of Helsinki working on the technology project New Methods and Applications for Speech Technology. He is also a student of Cognitive Technology at the Helsinki University of Technology, and his Diploma Thesis, to be completed in Spring 2005, concerns the MUMS multimodal navigation system. His primary research interests include natural interaction and interfaces, multimodal application development and modality fusion. E-mail: topi.hurtig@helsinki.fi

KRISTIINA JOKINEN is Professor of Language Technology at the University of Helsinki and a Visiting Fellow of Clare Hall at the University of Cambridge. Her research concerns AI-based spoken dialogue management, intelligent interactive systems, adaptive interfaces, rational agents, and natural cooperative communication. She developed the Constructive Dialogue Model approach for interaction management in dialogue systems and has directed several national and international research projects on spoken dialogue systems and adaptive user modelling. E-mail: kristiina.jokinen@helsinki.fi

THE CORPORA OF ESTONIAN AT THE UNIVERSITY OF TARTU: THE CURRENT SITUATION

Heiki-Jaan Kaalep, Kadri Muischnek
University of Tartu, Estonia

Abstract

This paper gives an overview of the corpus-related work done at the University of Tartu so far and describes an ongoing project – compiling a big corpus of written Estonian containing approximately 100 million words. The previously collected corpora of standard written Estonian at the University of Tartu are well-balanced and representative, but a little too small for the studies of statistically not so frequent phenomena in language, not to speak of the needs of language technology. The corpus under compilation right now, called the Mixed Corpus of Estonian, is planned as an open monitor corpus, but will also contain a more balanced subcorpus.

In addition to these corpora of standard written Estonian, the paper gives a very brief overview of the Corpus of Estonian Dialects, The Corpus of Spoken Estonian and the Corpus of Old Literary Estonian and discusses some special annotated corpora in more detail, namely the morphologically annotated corpus and the Estonian-English parallel corpus of legislative texts.

Keywords: Estonian language corpora, corpus linguistics, corpus compilation, corpus annotation

1. Introduction

In the recent years the focus of corpora-related work at the University of Tartu has been on building a big corpus of Estonian, consisting of at least 100 million words. A really big language corpus is essential for computational linguistics and for more theoretical branches of Estonian linguistics as well.

The easiest way to obtain written texts is to collect the texts that are already in an electronic form. We started from the texts available via Internet. Newspaper text is the dominating text type there, but one can find also legal texts, scientific texts, etc. We are trying to avoid manual work (downloading, converting, tagging) as much as possible. So we use special computer programs that do all this. Our final goal is to have a text that has been annotated up to the level of sentences, i.e. the headings, paragraphs, sentences and highlighted words/phrases are marked.

The work has been financed by the Ministry of Education via a national program "Eesti keel ja rahvuskultuur" (Estonian Language and National Culture).

2. The previous corpora at the University of Tartu

The history of corpus linguistics at the University of Tartu began in the first half of the nineties, when the 1-million word Corpus of Written Estonian was compiled. The work followed the well-known principles of Brown and Lancaster-Oslo/Bergen corpora: the corpus was divided into ten text classes that were designed to represent the whole written (edited and printed) culture from the years 1983-1988, the central year being 1985. This is a balanced sample corpus, each text excerpt containing maximally 2000 words. In addition to this, nine balanced subcorpora were compiled, one for each decade of the period 1890-1990, except for the 1980ies that were covered by the first corpus. These subcorpora contain about 300-400 thousand words each and they contain only two text classes: press and fiction - the largest text classes in Estonian culture and the only ones that exist through the 20th century in Estonian. The criteria of selection were the same as in the first corpus (for longer overview about selection of the fiction and press, see (Hennoste et al 1998)). Altogether this Thread of Corpora contains a little more than five million words. It gives a good overview of the development of the Estonian language during the 20th century, and has been in extensive use especially by students. But still it remains too small for studies addressing linguistic phenomena of lower frequency.

To make the picture complete, we should give a short overview of some other corpora being compiled at the University of Tartu as well.

The Corpus of Spoken Estonian[1] contains about 600 thousand words of transcribed speech, mostly everyday and institutional conversations. For more detailed description the reader is referred to (Hennoste et al 2001).

Closely related to the previous corpus is the Estonian Dialogue Corpus.[2] It contains three different types of dialogues: 1) spoken human-human dialogues, 2) written human-computer simulated interactions (using Wizard of Oz method), 3) human-computer interactions. The dialogues have been annotated for dialogue acts and communicative strategies. The reader will find a more detailed description of the dialogue corpus in (Koit 2002, Koit 2003).

The Corpus of Estonian Dialects[3] contains about 600 thousand words at the moment. The corpus contains texts in phonetic transcription, in simplified transcription, as well as morphologically annotated texts. One can use the corpus via Internet user interface. The reader will find a detailed description of the Corpus of Estonian Dialects in (Lindström, Pajusalu 2003).

The Corpus of Old Literary Estonian[4] contains over 700 thousand words and covers the period from the year 1224 up to the end of the 18th century. The corpus can be accessed via Internet user interface. For detailed description of the corpus the reader is referred to (Kingisepp et al 2004).

3. The Mixed Corpus of Estonian

3.1. The problem of representativeness and text classes

The ideal corpus of written language should represent all the text types that exist in the written culture of that language and the proportion of every text class in the whole corpus should correspond to the proportion of this text class in the whole body of written texts in a certain period. This is of course difficult to accomplish. In the history of corpus linguistics the well-known examples of well-balanced corpora are the Brown

[1] http://sys130.psych.ut.ee/~linds/

[2] http://www.cs.ut.ee/~koit/Dialoog/EDiC

[3] http://www.murre.ut.ee/

[4] http://www.murre.ut.ee/vakkur/

Corpus and the Lancaster-Oslo/Bergen Corpus. They contain only one million words each but have still remained valuable language resources. The British National Corpus that was compiled in the first half of the nineties “is a 100 million word collection of samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of current British English, both spoken and written.” as the homepage of BNC describes it.[5]

Most of the corpora of various languages are not balanced, mostly due to two reasons: 1) building an unbalanced corpus demands less resources and less time, and 2) deciding on the sublanguages and text classes that should be included and their proportions in the corpus is a difficult task and would doubtlessly be an object of severe criticism by the future users of the corpus.

This new Mixed Corpus of Estonian is planned as an open monitor corpus, meaning that it will not be necessarily balanced or representative. However, a smaller balanced subcorpus is also being compiled. It is described in more detail in section 3.3.

The new corpus contains only whole texts, no text samples. While the Thread of Corpora described in the previous section contained only texts initially written in Estonian, e.g. no translations, then for this corpus we collect also the translated texts. But there is one exception – we have decided not to include translated fiction or if included, it must be kept strictly separate from the fiction written in Estonian. The reason for this is the extremely bad translation quality of some fiction texts, especially in the text class of so-called commercial fiction (detective stories, love stories, etc.). For example the studies of word order based on these texts would show Estonian word order being very similar to that of English (but that is not the case, of course).

Like the Thread of Corpora, this new corpus will also contain no drama or poetry; it will contain only the written texts meant for reading – i.e. pre-planned and post-edited language usage mostly. But we will include more spontaneous and informal written speech, namely the language of newsgroups, internet forums and chatrooms.

The process of planning and collecting the corpus has revealed the fact that some text classes are underrepresented or totally absent in Estonian. For example it is quite difficult to find a scientific article in physics written in Estonian.

3. 2.The Current Situation

At the moment the Mixed Corpus contains the following subcorpora:

1) daily ‘Postimees’, 33 mio words
2) weekly ‘Eesti Ekspress’, 7,5 mio words
3) weekly ‘Maaleht’, 4,3 mio words
4) Estonian fiction, 4,2 mio words
5) PhD dissertations 500,000 words
6) popular science journal ‘Horisont’, 260,000 words
7) academic journal ‘Akadeemia’, 7 mio words
8) transcripts of Estonian Parliament (Riigikogu), 13 mio words
9) weekly “Kroonika”, 600,000 words
10) Estonian legislative documents, 1,8 mio words
11) Estonian translations of EU legislation, 9,6 mio words.

The newspaper text class is clearly overrepresented. We have two reasons for that. The first one is pragmatic: converting the newspaper texts into the corpus format

[5] http://www.natcorp.ox.ac.uk/

gave us maximal amount of words with minimal effort. But we also find the language used in newspapers being the closest to the so-called "general or standard Estonian".

3.3. The Balanced Corpus

To enable some comparative studies of the three (main) text classes of written Estonian, we have planned a balanced subcorpus within the Mixed Corpus. This will contain newspaper, fiction and scientific texts, five million words each. The newspaper part of it has been completed already, the fiction part we hope to complete soon, but the collecting and converting of the scientific texts still needs a considerable effort.

3. 4. Annotation

The general mark-up follows the TEI Guidelines.[6] The non-ascii characters are represented as SGML entities.

The division of the texts into paragraphs follows the original files. The headings and authors have been tagged. The text inside paragraphs has been processed by a program called estyhmm; as a result, the punctuation marks are separated from wordforms by a space (except those punctuation marks that are an integral part of the token, e.g. an abbreviation or an ordinal number) and the sentences are tagged with <s> and </s>. Every file starts with a header <teiHeader> documenting the file contents, size, used tags etc.

As for the markup of the initial structure of the text, the daily "Postimees" could serve as a nearly ideal example: the corpus has been divided into single newspaper issues, subdivisions of newspapers and newspaper articles, each of them being tagged as a division of separate level. As a result of this every sentence in the user interface can be linked to a source description giving the article, the author of the article, the subdivision and the newspaper issue were this particular sentence was printed.

4. Special subcorpora

In addition to these text collections our group has prepared some subcorpora with extra levels of annotation.

4.1. Morphologically disambiguated corpus

In this corpus[7] the text has been automatically analyzed by a program called estmorf (Kaalep, Vaino 2000) and subsequently manually disambiguated by two persons; and the third person has compared the result and made the necessary corrections.

The disambiguated texts belong to the following text classes:

Text class	*Number of tokens*
Fiction (Estonian authors)	104 000
G. Orwell's "1984"	75 500
Newspaper texts	111 000
Legal texts	121 000
Texts from a popular science journal "Horisont"	98 000
Reference texts	4 000
Altogether	513 000

[6] http://www.tei-c.org/Guidelines2/

[7] http://test.cl.ut.ee/korpused/morfkorpus/index.html.en

The word-forms have been analyzed one by one, except for some multi-word proper names like New York. The result of the analysis contains:

1) segmentation of the word into morphemes (stems and affixes)
2) lemmatization of the rightmost stem
3) syntactic word-class tag
4) morphological categories.

Ca 0,3% of the analyses can be debatable due to the ambiguousness of the borders between word classes or wrong because of human mistakes.

For more detailed description of the morphologically disambiguated corpus the reader is referred to (Kaalep et al 2000 or Muischnek, Vider 2005).

4.2. The Treebank of Estonian

The reader can learn about the Treebank (a corpus annotated for the phrase structure) of Estonian from (Uibo, Bick this volume).

4.3. The Estonian-English Parallel Corpus of Legislative Texts

This corpus contains:

1) Estonian-English parallel texts, 1.7 million tokens in Estonian, 2.9 million tokens in English.
2) English-Estonian parallel texts, 3.3 million tokens in Estonian, 4.9 million tokens in English.

The texts originate from Estonian Legal Language Centre[8] on April 30, 2002. The aligned versions are based on the TEI P3 compatible versions of the same files from the Mixed Corpus of Estonian.

The texts have been sentence-aligned. The items of lists are treated as equal to sentences. The Estonian and English sentences may be in 1-1, 1-2 or 2-1 alignments. There are no other alignments (like 1-0, 0-1, 2-2 etc) in this corpus. They were either not found or they were left aside as they would be hard to use in future work, the aim of which is to find parallel multi-word units. The aligning was done using the Vanilla aligner.[9] It is a language independent aligner, based on the algorithm from (Gale, Church 1993).

5. User interface

The Mixed Corpus of Estonian and the Corpus of Written Estonian could be used via Internet interface.[10]

All the texts are divided into sentences, so one always gets a full sentence as an answer to her/his query. It is also possible to ask for up to five preceding and following sentences and so get more contextual information. It is only possible to seek for a word-form (or string including regular expressions) as these corpora have not (yet) been morphologically annotated. At the moment all the texts in the user interface are represented as plain text, all tags removed.

The morphologically annotated corpus described in 4.1. can be accessed via its own interface.[11]

[8] http://www.legaltext.ee
[9] http://nl.ijs.si/telri/Vanilla/
[10] http://test.cl.ut.ee/korpused/kasutajaliides/index.html.en
[11] http://test.cl.ut.ee/korpused/morfliides/index.html.en

References

Gale, W. A.; Church, K. W. 1993. Program for aligning sentences in bilingual corpora. In: *Computational Linguistics* 19, 75–102.

Hennoste, Tiit; Roosmaa, Tiit; Saluveer, Madis 1998. Structure and usage of the Tartu University Corpus of Written Estonian. In: *International Journal of Corpus Linguistics* 3 (2), 279–304.

Hennoste, Tiit, Lindström, Liina, Rääbis, Andriela, Toomet, Piret, Vellerind, Riina 2001. Tartu University Corpus of Spoken Estonian. In: *Congressus Nonus Fenno-Ugristarum Pars V*. Tartu, 345–351.

Kaalep, Heiki-Jaan; Muischnek, Kadri; Müürisep, Kaili; Rääbis, Andriela; Habicht, Külli 2000. Kas tegelik tekst allub eesti keele morfoloogilistele kirjeldustele? Eesti kirjakeele testkorpuse morfosüntaktilise märgendamise kogemusest. In: *Keel ja Kirjandus* 9, 623–633.

Kaalep, Heiki-Jaan; Vaino, Tarmo 2001. Complete Morphological Analysis in the Linguist's Toolbox. In: *Congressus Nonus Internationalis Fenno-Ugristarum Pars V*. Tartu, 9–16.

Kingisepp, Valve-Liivi; Prillop, Külli; Habicht, Külli 2004. Eesti vana kirjakeele korpus: mis tehtud, mis teoksil. In: *Keel ja Kirjandus* 4, 272–283.

Koit, Mare 2002. Kommunikativnye strategii v informacionno-spravochnom dialoge (na materiale estonskogo korpusa dialogov). In: *Proc. DIALOG-2002, 6-11 June 2002 b.* Vol. 2, Moskva: Nauka, 283–290.

Koit, Mare 2003. Märgendatud dialoogikorpus kui keeleressurss. In: *Toimiv keel I. Töid rakenduslingvistika alalt.* Eesti Keele Instituudi toimetised 12. Tallinn: Eesti Keele Sihtasutus, 119–136.

Lindström, Liina; Pajusalu, Karl 2003. Corpus of Estonian Dialects and the Estonian vowel system. In: *Linguistica Uralica* 4, pp 241–257.

Muischnek, Kadri; Vider, Kadri 2005 (to appear). Sõnaliigituse kitsaskohad eesti keele arvutianalüüsis.

Uibo, Heli; Bick, Eckhard 2005. Treebank-based research and e-learning of Estonian syntax. (*This volume.*)

Viks, Ülle 1992.Väike vormisõnastik I. Sissejuhatus & grammatika; II. Sõnastik & lisad. Tallinn.

HEIKI-JAAN KAALEP is a senior researcher at the Working Group of Computational Linguistics, University of Tartu. A description of his work and list of (also on-line) publications one can find at his homepage http://test.cl.ut.ee/inimesed/hkaalep/index.html.en

KADRI MUISCHNEK is a researcher at the Working Group of Computational Linguistics, University of Tartu. For details see also http://test.cl.ut.ee/inimesed/kmuis/index.html.en

TEKSAURUS — THE ESTONIAN WORDNET ONLINE

Neeme Kahusk, Kadri Vider
Tallinn Pedagogical University, University of Tartu, Estonia

Abstract

TEKsaurus — the Estonian WordNet Online — makes use of Estonian part of concept-based lexical database EuroWordNet. The tesaurus is frequently updated and supplied with all semantic relations found in Estonian WordNet, and Princeton WordNet equivalents. The online version is based on new Python modules for parsing EuroWordNet export files.

Word senses in TEKsaurus are also linked with Word Sense Disambiguation Corpus of Estonian (WSDCEst) and disambiguated word senses in WSDCEst are linked with TEKsaurus. The paper gives also a brief overview of TEKsaurus statistics.

Keywords: EstWN, EuroWordNet, Python, EKSS, word frequency, Filosoft thesaurus

1. Introduction

The beginning of Estonian WordNet (EstWN) was in 1996. In 1998 the Estonian team joined EuroWordNet, a project financed by European Comission. After finish of the EuroWordNet, EstWN is continued on its own, and as a by-product of word sense disambiguation (WSD) task.

Since 2002 EstWN is available to public as an online query. At first it was one part of a lexicographer's tool for WSD (Kahusk 2002). Starting from 2004 EstWN is available as a separate service under name TEKsaurus[1]. The server is running a cgi script written in Python 2.2.

Having been part of EuroWordNet, the Estonian WordNet is built according to the same principles and in the same format. The number of synsets, literals and semantic relations have grown since the EuroWordNet deliverables were issued — the Estonian part is described in Vider et al. (1999). By now, there are 10,884 synsets in EstWN, 14,977 lexical entries and 23,426 semantic relations.

For a long time, the only possibility to explore Estonian Wordnet was to use the Polaris database tool, or Periscope viewer, that is intended to browse wordnet, and not to edit it. Polaris is meant to be the main tool for editing EuroWordnet. It is a powerful tool and has a good GUI. Still, Polaris has some weak points. It is a

[1] http://www.cl.ut.ee/ → Resources → TEKsaurus

proprietary software and is no longer supported. By our knowledge Periscope has never gained any popularity as an EuroWordNet browsing tool.

The EuroWordNet (Polaris) import-export format is specified in Louw (1997). The format is based on GEDCOM standard. EuroWordNet record is a hierarchically organised collection of fields, forming a tree. The syntax of data lines is very simple: each line consists of level number, optional database record number (this is present only in level 0 lines), field tag and optional field value. The maximum level number can be 31.

The import-export format is simple enough for some editing tasks. Still, the inner structure of a wordnet is complicated, and especially linking of semantic relations is not a reasonable job to do on plain text file. We have continued to use the Polaris tool for editing, and for browsing we have worked out the online version of EstWN — the TEKsaurus.

2. Browsing TEKsaurus

The main ideas of online GUI of TEKsaurus is simple design and ease of usage. As there are no morphological tools integrated, the queries should be in base (dictionary keyword) form — singular nominative for nouns, and adjectives and *ma*-infinitive for verbs.

The dictionary keywords were the only possibility for queries in the previous version of the online EstWN. The present version provides the means to find synsets directly. One can enter explicitly the numerical id of the synset — preceeding with #, or provide sense number and part of speech with query, separated with commas.

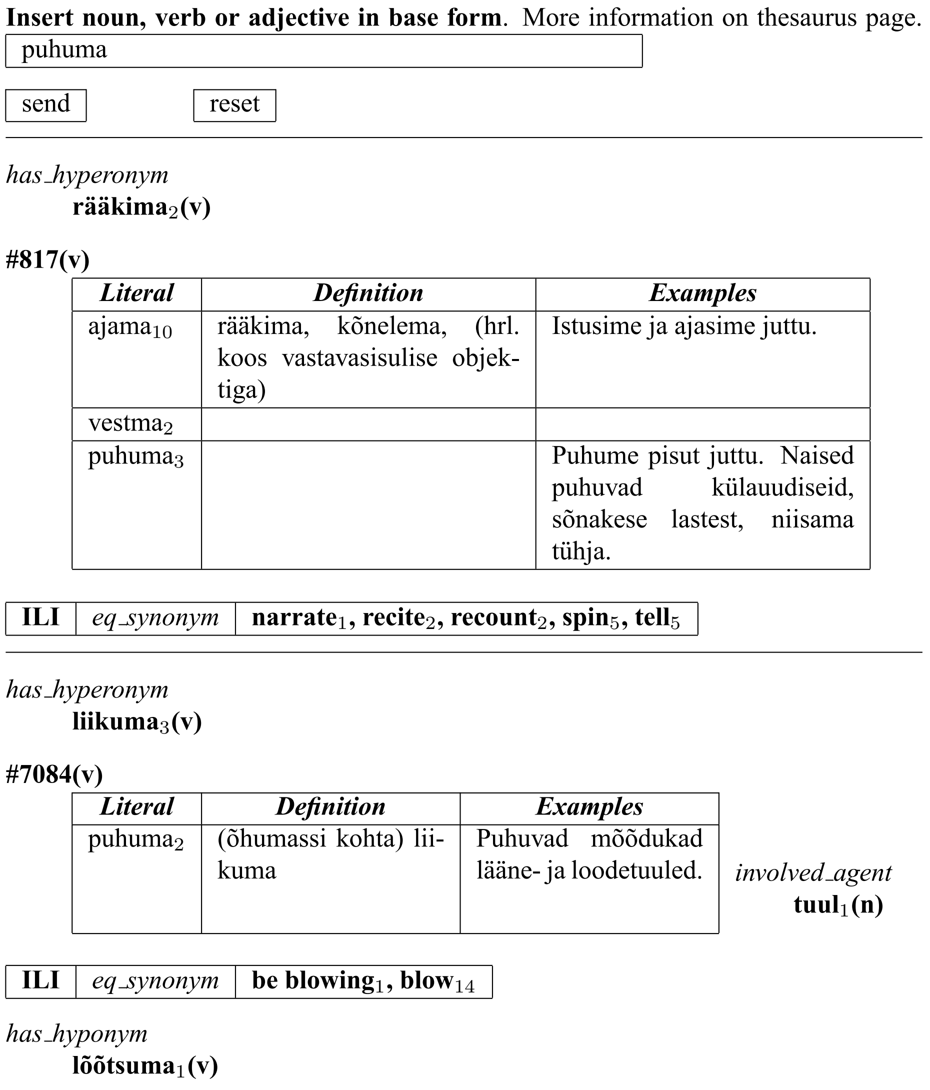
The answer to the query is grouped by synsets. According to the idea of wordnet, the answer can be many synsets, and mostly it is the case. See Table 1 for distribution of synsets according to number of members.

For every synset the answer provides a table with synset data: synset number, literal(s), definition, and examples. Besides this, a table with ILI links, is presented, and semantic relations in three categories: (1) hyperonym(s) (or *belongs_to_class*), (2) hyponyms (or *has_instance*), and (3) other relations. See Figure 1.

All the lexical items are clickable: every literal (member of synset) can serve as the source for new query as a keyword. Semantic relation terminals are clickable as well, but they led to a synset query, as they provide sense number and part of speech too.

Table 1: Distribution of synsets by number of synset members

No. of synsets	% of total synsets	No. of synset members
5509	50.6	1
2869	26.4	2
1405	12.9	3
506	4.7	4
149	1.4	5–9



Insert noun, verb or adjective in base form. More information on thesaurus page.

puhuma

send reset

---

has_hyperonym
rääkima$_2$(v)

#817(v)

Literal	*Definition*	*Examples*
ajama$_{10}$	rääkima, kõnelema, (hrl. koos vastavasisulise objektiga)	Istusime ja ajasime juttu.
vestma$_2$		
puhuma$_3$		Puhume pisut juttu. Naised puhuvad külauudiseid, sõnakese lastest, niisama tühja.

ILI	*eq_synonym*	**narrate$_1$, recite$_2$, recount$_2$, spin$_5$, tell$_5$**

---

has_hyperonym
liikuma$_3$(v)

#7084(v)

Literal	*Definition*	*Examples*
puhuma$_2$	(õhumassi kohta) liikuma	Puhuvad mõõdukad lääne- ja loodetuuled.

involved_agent
tuul$_1$(n)

ILI	*eq_synonym*	**be blowing$_1$, blow$_{14}$**

has_hyponym
lõõtsuma$_1$(v)

Figure 1: Teksaurus answer to query ‘puhuma’. Above first line, there is the query part. The answer is found from two synsets: #817, translated as ‘to narrate, to tell’, and #7084, translated as ‘be blowing, to blow’. ‘puhuma$_3$’ has a hyperonym only, ‘puhuma$_2$’ has a hyperonym, one hyponym and involved agent ‘tuul$_1$’ (*wind*)

Table 2: Main figures characteristic to EstWN

	noun	verb	adjective	proper noun	total
synsets	7,336	2,795	307	446	10,884
word senses	12,067	5,777	518	471	18,833
lexical entries (lemmas)	10,287	3,801	419	470	14,977
semantic relations	16,463	5,951	539	473	23,426
relations per synset	2.24	2.13	1.76	1.06	2.15
senses per synset	1.64	2.07	1.69	1.06	1.73
senses per lexical entry	1.17	1.52	1.24	1.0	1.26

Table 3: Comparison of keywords found in EstWN (TEKsaurus) and Explanatory Dictionary of Estonian (EKSS)

	TEKsaurus			EKSS (A-sulforühm)		
	single units	phrases	total	single units	phrases	total
total no. of keywords	11,525	986	12,511	98,990	3,990	102,980
unique keywords (not in other resource)	1,761	839	2,600	89,226	3,843	93,069
intersection	9,764	147	9,911	9,764	147	9,911

3. Behind the scene

The engine behind TEKsaurus is a Python script running on server. In first stage, the EWN export file is used to generate two index files and pickled object files for every synset. At first, there are generated two indexes: the literal index and synset index. The literal index is a Python dict array, which has all different literals as keys. A list corresponds to every key, there are the synset id numbers, where the literal can be found.

4. EstWN coverage and lexical resources

The Frequency Dictionary of Written Estonian (Kaalep and Muischnek 2002) is compiled on the basis of the same corpus what we use in WSD. There are 6810 single words in this dictionary that belong to verb or noun classes, 5071 of them are in EstWN too. That seems to be pretty good coverage. We consider frequency of five occurrences the threshold to add a word into the thesaurus.

There are plenty of dictionaries that have served as lexical resources for EstWN. The most prominent source has been the Explanatory Dictionary of Estonian (EKSS 1988) that is still in progress. There is an electronic version of the dictionary as well. The version we have reached is not a machine-readable dictionary in its best sense, but a collection of inconsistently tagged text files. Needless to say, that electronic versions are preferrable: at least one can do some simple search and sort of items. The other electronic dictionary we have used is the Filosoft thesaurus, that is based on Estonian dictionary of synonyms (Õim 1991).

The number of lexical entries in EKSS is abut ten times more than in TEKsaurus. Still, there are more than 1000 TEKsaurus words, that can not be found from EKSS. Most of them (about 85 %) are nouns, mostly compounds and deverbal infinitives.

The Filosoft thesaurus (FS) lacks word meanings without synonyms. In EstWN, there are about half of synsets with one member (Table 1). The other synsets (4929) are compatible with FS, and 1428 of them have injective mapping. There are 4866 synonym rows in FS, that have intersection of members with EstWN. But still, we have a close aim to get, as there are 4414 synonym rows in FS that completely miss from EstWN.

5. Acknowledgements

This paper is based on work supported in part by the Estonian Science Foundation under grant No 5534 “Concept based resources and processing tools for the Estonian language” and by Estonian State Target Financing R&D project No 0182541s03 “Computational models and language resources for Estonian: theoretical and applicational aspects” and by Governmental programme “Eesti keel ja rahvuslik mälu” subproject “Language Technology: Semantic analysis of Estonian simple sentence”.

References

EKSS 1988. Eesti Kirjakeele Seletussõnaraamat. Explanatory Dictionary of Estonian. Publishing is in progress

Kaalep, Heiki; Muischnek, Kadri 2002. Eesti kirjakeele sagedussõnastik. Tartu. In Estonian. English title: The Frequency Dictionary of Written Estonian

Kahusk, Neeme 2002. A lexicographer’s tool for word sense tagging according to WordNet. In: Christodulakis, D. N.; Kunze, C.; Lemnitzer, L. (eds.), *Workshop on Wordnet Structures and Standardisation, and how these Affect Wordnet Applications and Evaluation*. 1–7

Louw, Michael 1997. EuroWordNet Import Specifications. Paper 010. EuroWordNet, LE2–4003

Õim, A. 1991. Sünonüümisõnastik. Tallinn. (Estonian dictionary of synonyms)

Vider, Kadri; Paldre, Leho; Orav, Heili; Õim, Haldur 1999. The estonian wordnet. In: Kunze, C. (ed.), *Final Wordnets for German, French, Estonian and Czech*. EuroWordNet (LE-8328), Deliverable 2D014

KADRI VIDER is researcher and PhD student at Department of General Linguistics, University of Tartu. She received her M.A. in 1999 dealing with senses of Estonian verbs in semantic database such as wordnet. Her research interests concern computational lexicology, lexical semantics and word sense disambiguation. Her doctoral study focuses on senses of Estonian verbs

and possibilities to distinguish them in texts. She is member of the board of the Estonian Association of Applied Linguistics. E-mail: kadri.vider@ut.ee

NEEME KAHUSK is reseacher at Department of General Linguistics, University of Tartu, and PhD student at Department of Psycholoogy, Tallinn Pedagogical University. He received his M.Sc in psychology in 2002. His main research interests are in computational lexicology, lexical semantics and psycholinguistics of word explanations. His PhD study focuses on word explanations given at limited time conditions. E-mail: neeme.kahusk@ut.ee

AUTOMATIC ANNOTATION OF SENTENCE BOUNDARIES AND CONTRACTIONS IN LITHUANIAN TEXTS

Jurgita Kapočiūtė, Gailius Raškinis
Vytautas Magnus University (Kaunas, Lithuania)

Abstract

In this paper we present an algorithm that detects, recognizes and annotates sentence boundaries and contractions in Lithuanian texts. The algorithm is based on a set of template matching rules that include matching of a surface text against external linguistic knowledge. The annotation is performed in multiple passes over the text. During the first pass all text tokens are tested for being possible word forms of standard Lithuanian, proper nouns or word forms of a foreign origin. During the second pass, contractions are recognized and annotated. During the final pass, sentence boundaries are detected and annotated. The algorithm can operate in automatic or semi-automatic annotation modes. The semi-automatic mode allows users to intervene in cases where automatic decision is uncertain. Users' feedback in memorized and the external linguistic knowledge is extended. The algorithm is evaluated on an expert-annotated corpus of 300 000 words and resulted in 99% accurate annotations.

Keywords: text corpora, SGML annotation, sentence boundaries, contractions.

1. Introduction

Text corpora are large text collections that store many millions of running words. The main purpose of a corpus is to provide a basis for verification of hypothesis about language. The verification of sophisticated hypothesis requires annotated corpora, i.e. corpora to which additional linguistic, morphological, and syntactic information has been added. Though manually created text annotations can be very accurate, manual annotation is extremely human labor-intensive.

This paper partially describes our ongoing work aiming at methods for automatic annotation of Lithuanian text corpora. In particular, we are looking for methods that can annotate sentence boundaries and contractions, i.e. for methods that solve period (".") related ambiguity in Lithuanian texts. Automatic detection of sentence boundaries and automatic recognition of contractions is important for language modeling, speech synthesis, morphological disambiguation, parallel corpora compilation, monitoring of written language evolution and other purposes.

2. Related work

The methods of automatic detection of sentence boundaries usually fall into two broad categories. Either empirically stated template matching rules or probabilistic models estimated from annotated text corpora are used.

Simple template matching rules are described by Wang at al. (2003). Authors state that a sentence boundary can be detected by scanning text for the sequence of a lower-case string, one of the symbols “.”, “?”, “!”, and a capital letter. More sophisticated template matching rules are proposed by Kiss et al. (2003). Authors use a parameterized rule, where parameters are extracted from annotated text corpora. The rule is based on features such as: length of word, number of internal periods it has (e.g., “U.S.A.” has two internal periods), number of times each word goes at the end of the sentence in training corpus, how often each word begins with a capital or lower-case letter; what words most frequently go at the beginning of the sentence. Candidate pattern is classified either as being a contraction or a sentence boundary depending on whether the rule applied to the candidate pattern results in a value exceeding specified threshold. Kiss et al. applied their method to 8 Indo-European languages as well as to Estonian and Turkish and reported 98.93% – 99.72% and 90.52% – 99.92% annotation accuracy for sentence boundaries and contractions respectively.

Wang et al. (2003) are addressing sentence boundary detection problem within a probabilistic framework. The authors compare Hidden Markov Models (HMM) trained on an annotated text corpus and the maximum entropy approach. In the latter case, word collocations and their frequencies at the beginning and at the end of the sentence are used as features and integrated into the formula of maximum entropy that is used for identifying sentence boundaries in the text. Tajima et al. (2003) uses similar probabilistic methods for identifying sentence boundaries: phrases are analyzed, examining how often certain words, phrases or morphological tags can go at the end of the sentence. Accuracy of annotating English sentences reported by Wang et al. (2003) varies from 91.43% to 99.56%. Tajima et al. (2003) report 77,24% accuracy for Japanese texts.

Automatic boundary detection of Lithuanian sentences has been never attempted. In this paper we present the first automatic annotation algorithm of Lithuanian sentences and contractions based on a template matching approach.

3. Automatic annotation

3.1. Annotation problems

Automatic detection and discrimination of sentence boundaries from contractions is closely related to the following problems:

Compound contractions. The contractions can be simple or compound. Compound contractions may have one or several internal periods and one external period while simple contractions haven’t any internal periods.

Notational variation of contractions. The notation of Lithuanian contractions varies a lot. Contractions may start both with capital and lower-case letter, they can end with or without an external period. There are contractions that aren’t short at all.

Sentence ending with a contraction. The contraction that ends with an external period can be followed by a proper noun (always beginning with the capital letter). This is an ambiguous case very similar to the sentence boundary on the surface level.

3.2. Proposed algorithm

The algorithm described in this paper uses a template matching approach. The templates often refer to an external linguistic knowledge stored in databases. For instance, “is the text token a word form of standard Lithuanian?”, “does the text token belong to the list of known contractions?” are a few pieces of knowledge the templates may require. The algorithm can operate in automatic and semi-automatic processing modes. When operating in a semi-automatic processing mode, the algorithm may refer to a human to provide the correct boundary/contraction decision. This happens if database lookup fails and its own processing templates cannot achieve required certainty. Human's answers are always stored in one of the external databases, thus extending linguistic knowledge and reducing the number of appeals for human help in future (Figure 1).

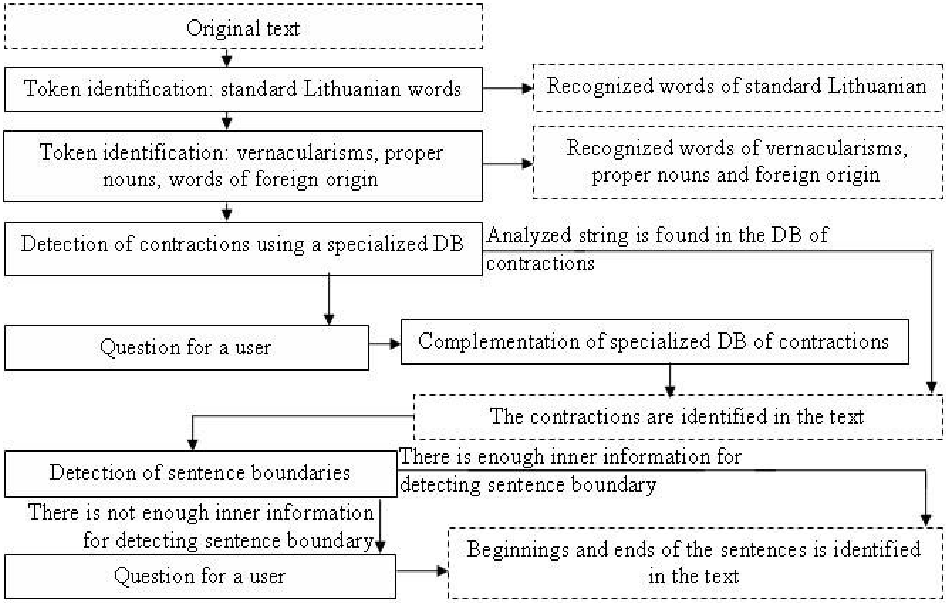

Original text
Token identification: standard Lithuanian words
Recognized words of standard Lithuanian
Token identification: vernacularisms, proper nouns, words of foreign origin
Recognized words of vernacularisms, proper nouns and foreign origin
Detection of contractions using a specialized DB
Analyzed string is found in the DB of contractions
Question for a user
Complementation of specialized DB of contractions
The contractions are identified in the text
Detection of sentence boundaries
There is enough inner information for detecting sentence boundary
There is not enough inner information for detecting sentence boundary
Question for a user
Beginnings and ends of the sentences is identified in the text

Figure 1. The architecture of an algorithm for annotating contractions and sentence boundaries

The algorithm of automatic text annotation consists of three main stages:

Token identification stage. During this stage, text tokens are tested for being possible word forms of standard Lithuanian[1]. Tokens appearing not to be word forms of standard Lithuanian are tested for being vernacularisms, proper nouns and words of foreign origin. Candidate vernacularisms are recognized by the simple replace-match rule. They have their typical endings ("on", "oj", "im", etc.) replaced by appropriate standard Lithuanian endings (e.g., "on" -> "a" "rankon" -> "ranka") and are tested for being possible word forms of standard Lithuanian repeatedly. Proper nouns and words of foreign origin are recognized by looking up into the specialized linguistic databases. Finally, a paragraph numeration check is performed: a regularly repeated numeration (numerical or literal) is identified.

Contraction identification stage. During this stage, the remaining unidentified tokens are matched against the entries of a database of simple and compound contractions. Sometimes adjacent tokens constitute two neighboring contractions: compound and simple. In such a case, contractions should be annotated separately. If text token has internal periods and if it is already a contraction candidate, the algorithm (working in the semi-automatic mode) asks the user if this sequence of tokens is the

[1] Text token is identified as a word form of standard Lithuanian if it is recognized by the morphological lemmatizer of Lithuanian (Zinkevičius, 2000)

compound contraction or it contains several simple ones. Other text fragments (even if they don’t have internal or external periods) are checked and identified the same way. During this stage, the information about new contractions is collected and incorporated into the contractions database.

Sentence boundaries identification stage. Detection of sentence boundaries is performed by means of a set of empiric template matching rules. Let:

$L be the set of lower-case letters;
$U be the set of upper-case letters;
$D be the set of digit characters;
$S be the set of token separators [,.?!:-/()[]{}<>|%`"-'˙–*];
$M be the set of usual sentence separators [.!?];
$C be the set of known contractions[2] never occurring at the end of a sentence;
$A be the set of known contractions occurring anywhere in the text.
$P be the set of known personal names;
$R be the set of paragraphs numeration symbols (ending with external period);
$W be the set of proper nouns;
$T denote the set of words of standard Lithuanian;
\A denote[3] the beginning of a new line;
\Z denote the end of line.

Let X{nmin,nmax} denote the fact that the symbol X is allowed to be adjacently repeated from to nmin to nmax times. Let <s> and </s> indicate the positions of true annotation candidates of sentence start and end respectively. Let <–> indicate the position of a false annotation candidate. Let <?> denote the position of an annotation candidate which should be classified by a human if operating in the semi-automatic mode and which should be held equivalent to <–> in fully automatic mode. The sentence boundary identification is performed by the set of following template matching rules:

1. \A<s>$S{0,}$U{1,}[4]
2. \A<s>$S{0,}$L{1,}
3. \A<s>$S{0,}$D{1,}
4. \A<–>$S{1,}$U{0}
5. \A<–>$S{1,}$L{0}
6. \A<–>$S{1,}$D{0}
7. $U{1,}$S{0,} </s>\Z
8. $L{1,}$S{0,} </s>\Z
9. $D{1,}$S{0,} </s>\Z
10. $U{0}$S{1,} <–>\Z
11. $L{0}$S{1,} <–>\Z
12. $D{0}$S{1,} <–>\Z
13. $L{1,}$S{0,}$M{1,}</s><s>$S{0,}$U{1,}
14. U{1 }<–>P{1 }
15. $R{1}<–>$U{1,}
16. $R{1}<–>$D{1,}
17. $C{1}<–>$U{1,}
18. $C{1}<–>$D{1,}
19. $A{1}</s><s>$T{1}
20. $A{1}<?>W{1}
21. $D{1,}$S{0,}$M{1,}<?>$S{0,}$U{1,}
22. $L{1,} $S{0,}$M{1,}<?>$S{0,}$D{1,}
23. $U{1,} $S{0,}$M{1,}<?>$S{0,}$D{1,}
24. $D{1,} $S{0,}$M{1,}<?>$S{0,}$D{1,}

Finally, sentences and contractions are annotated by SGML (Standard General Markup Language) tags.

[2] All sets of contractions will have external periods.
[3] This is a Perl inspired notation.
[4] $U can be repeated an infinity of times but at least once.

4. Results

The methods developed in this paper is tested on and to evaluate the precision of annotating while comparing an expert (annotated manually) text and annotated while using an algorithm.

The algorithm described in this paper was tested in both automatic and semi-automatic operation modes. Expert-annotated texts were compared with the automatically annotated texts on a tag-by-tag basis. Texts consisting of 300 thousand words[5] were used for testing. The annotation accuracy A was calculated using the formula:

$$A = \frac{N - I - D}{N} \times 100\%$$

where N, I, and D denote correct, false (inserted), and missed (deleted) annotations respectively (Table 1).

Table 1. Accuracy annotating sentence boundaries and contractions

Annotation type	Processing mode	
	Automatic	Semi-automatic
Sentence boundaries	98,0%	99,2%
Contractions	98,5%	99,5%

Semi-automatic processing mode resulted in approximately 50 questions per thousand tokens.

Annotation errors appeared to be dues to the following reasons:

Spelling mistakes. The algorithm assumes the absence of spelling mistakes in a text. Misspelled contractions, for example, fail to be annotated.

Users' failures. When operating in a semi-automatic annotation mode, the algorithm assumes human input is error-free. Dubious annotation cases resolved by humans are stored within linguistic databases and used for further enhancement of automatic annotation. Thus, erroneous human input harms the accuracy of automatic annotation.

Imperfect templates. The templates used by our algorithm are not perfect. Some sentences do not end with the usual punctuation marks (".", "?", "!"). Some cases require deeper contextual analysis. For example, as quotation marks can identify the end of the sentence, the boundaries of some sentences can be detected only after conducting an analysis of opening and closing of quotation marks.

Absence of learning. The templates were stated after an empiric investigation. Annotating previously unseen text cases of sentence boundaries emerge, for which there are no templates present.

5. Conclusions

This paper presents an algorithm for automatic annotation of sentence boundaries and contractions. The precision of the proposed algorithm is evaluated by comparing machine annotated texts with texts annotated by a human expert. The algorithm reaches over 98% precision working in completely automatic mode.

[5] Texts were taken from the 100 million word text corpus compiled at Vytautas Magnus University (Marcinkevičienė, 2000).

The algorithm is adaptive in the sense, that in semi-automatic operating mode human's answers are always stored in one of the external databases. Thus linguistic knowledge becomes extended and the number of future appeals reduced.

The processing architecture used for annotating sentence boundaries and contractions can be extended for detection and annotation of other linguistically distinct text elements: dates, spelling mistakes, proper nouns and others.

References

Kiss T., Strunk J. 2003. Multilingual least effort sentence boundary detection, from http://www.linguistics.ruhr-uni-bochum.de/~strunk/ks2003FINAL.pdf.

Marcinkevičienė R. 2000. Corpus linguistics in theory and practice. In: *Darbai ir Dienos, VDU* 24. Retrieved December 12, 2000, from http://donelaitis.vdu.lt/publikacijos/marcinkeviciene.pdf.

Tajima S., Nanba H., Okumura M. 2003. Detecting sentence boundaries in Japanese speech transcriptions using a orphological analyzer, from http://www.nlp.its.hiroshima-cu.ac.jp/~nanba/pdf/ijcnlp_tajima.pdf.

Wang H., Huang Y. 2003. Bondec – a sentence boundary detector, from http://nlp.stanford.edu/courses/cs224n/2003/fp/huangy/final_project.doc.

Zinkevičius V, 2000. Lemuoklis – tool for morphological analysis. In: *Darbai ir Dienos, VDU* 24. Retrieved December 12, 2000, from http://donelaitis.vdu.lt/publikacijos/zinkevicius.pdf.

Electronic text center. Retrieved January 18, 2005, from http://etext.lib.virginia.edu.

TEI-Text Encoding Initiative. Retrieved March 7, 2002, from http://helmer.aksis.uib.no/tonemerete/forelesninger/Datalingvistikk/om_tei_2002_03_07.html.

JURGITA KAPOČIŪTĖ is postgraduate student in computer science at Vytautas Magnus University in Kaunas, Lithuania. She received her B.Sc in Computer Science at Vytautas Magnus University in 2003. Her research interests are related to automated text annotation. E-mail: Jurgita_Kapociute@fc.vdu.lt

GAILIUS RAŠKINIS received his M.Sc. degree in artificial intelligence and pattern recognition from the University of Pierre et Marie Curie in Paris in 1995. He received Doctor's degree in the field of informatics (physical sciences) in 2000. Presently, he works at the Center of Computational Linguistics and teaches at the Department of Applied Informatics of VMU. His research interests include application of machine learning techniques to human language processing. E-mail: g.raskinis@if.vdu.lt.

WHERE DO CONCEPTUAL SPACES COME FROM? AN EXAMPLE OF THE ESTONIAN EMOTION CONCEPTS

Toomas Kirt*, Ene Vainik**
*Tallinn University of Technology, Tallinn, Estonia
**Institute of the Estonian Language, Tallinn, Estonia

Abstract

The key question in cognitive science is how to represent concepts. In this paper, we present results of our study of the Estonian emotions concepts in the light of the theory of conceptual spaces. The purpose of our study is to find out if there is an underlying universal structure of emotion knowledge that is independent of the nature of the source data and analytical tools. In the empirical study we report the results of 100 Estonian subjects. As an analytical tool we used the self-organizing maps (SOM) that is a useful method to classify and visualise multidimensional data. Another benefit of the self-organising maps is that it simulates partly the self-organizing processes that take place in the human brain. It converts the nonlinear statistical relationships between high-dimensional data into simple geometric relationships of their image points on a regular two-dimensional grid of nodes. The SOM is useful tool for identifying quality dimensions that the conceptual space is based on.

Keywords: self-organizing maps, neural networks, semantics, linguistics, conceptual spaces

1. Introduction

The idea, that human conceptual representation of the world – or the mental lexicon (Aitchison 2003) – is structured and organized by nature, not an arbitrary mess of words is widespread throughout the cognitive linguistics (e. g. Langacker 1987, Viberg 1994 & Cruse 2000 among others). This presupposition should hold in all cognitive domains, including the culturally shared knowledge about mental life – e.g. emotions. On the other hand, the statistical studies carried out in the field of psychology relying on the results of different lexical tasks tend to end up with controversial solutions as regards to the structure of the emotion lexicon (see Russell 1980, Watson & Tellegen 1985 for example). The organizing "dimensions" of some semantic field spoken intuitively about in the cognitive linguistics seem not to match with the results of factor analysis or multidimensional scaling applied on the empirical data of that particular field, nor do the last match each other.

The purpose of present study is to find out if there is an underlying universal structure of emotion knowledge that is independent of the nature of the source data and analytical tools (see Vainik 2004 for a more details). As a theoretical framework for this paper we chose the P. Gärdenfors's theory of conceptual spaces (2000a & 2000b) that is compatible with the idea of neural networks and self-organization as the two general principles used in both computational and human data processing. In the following paper we use the empirical data of the Estonian emption terms to illustrate the process of self-organization and to discuss where do the conceptual spaces (or the inner structure of a semantic field) come from.

2. The Main Ideas of the Theory of Conceptual Spaces

Gärdenfors (2000) proposed a geometrical model for representations of concepts called Conceptual Spaces. In this model he distinguishes three levels of representations: the most abstract level is the symbolic level on which the observation is described by means of some language, the second level is the conceptual level on which observations are located as points in the conceptual space and the least abstract level is the subconceptual level on which the observations are characterized by inputs from sensory receptors which form the dimensions of the conceptual space.

A conceptual space is described by a number of quality dimensions. Qualities are mostly measured by our sensory receptors, but they can be more abstract by nature as well. Quality dimensions describe the properties of an object and relations between the properties. Quality dimensions are divided into two groups: integral and separable. A value on an integral dimension is always co occurring with another value measurable on another dimension: for example the hue and brightness of an object are inseparable, integral dimensions. Independent dimensions on the contrary are separable in principle, for example the size and hue of an object are considered as independent dimensions. The function of the quality dimensions is to represent the "qualities" of the observations and to build up domains needed for representing concepts. Spatial dimensions belong to one domain, colour dimensions to another and so on. The notion of a cognitive domain can be defined as a set of integral dimensions that are separable from all the other dimensions.

The quality dimensions are the main tool for measuring similarity of the concepts. If we assume that dimensions are metric then we can talk about distances in the conceptual space. The smaller the distance is between the representations of two objects, the more similar they are. In this way, the similarity of two objects can be defined as the distance between their representing points in the space.

A conceptual space can be defined as a collection of one or more domains. A point in the space may denote a concept. The properties of the object can be identified with its location in space. And a property can be represented as a region of the domain. The domains of a conceptual space should not be seen as totally independent entities, but they are correlated in various ways since the properties of the objects modelled in the space covary. In symbolic level we can say that "all A-s are B-s" and in conceptual level it means that there is a strong correlation between an object in conceptual space and a certain value of its property.

As Gärdenfors has proposed there is an analogy between the Conceptual Spaces and the Self-Organizing Maps. During the self-organizing process the points in high-dimensional space are mapped onto a two-dimensional output map that can be identified as a Conceptual Space. The self-organizing map is one way of modelling how the geometric structure within a domain can be created from the information on the subconceptual level.

3. Method: The Self-Organizing Maps

The self-organising map is a feedforward neural network that uses an unsupervised training algorithm (Kohonen 2000, Deboeck & Kohonen 1998). The algorithm provides a topology- preserving mapping from high-dimensional space to map units. Map units, or neurones, usually form a two-dimensional space grid and thus the mapping is a mapping from a high-dimensional space onto a plain. The property of topology preserving means that the SOM groups data vectors of similar input on neurons: the points that are near each other in the input space are mapped to nearby map units in the SOM. The SOM can thus serve as a clustering tool as well as a tool for visualising high-dimensional data. The process of creating a self-organising map requires two layers of processing units.

The learning process goes on as follows. At first the output grid will be initialised with initial values that could be random values from the input space. One sample will be taken from the input variables and presented to the output grid of the map. All the neurons in the output layer compete with each other to become a winner. The winner will be the output node that is the closest to the sample vector. The distance between two vectors is measured by Euclidean distance. The weights of the winner neuron will be changed closer to the sample vector, moved in the direction of the input sample. The weights of the neurons in the neighbourhood of the winner unit will also be changed. During the process of learning the learning rate becomes smaller and the rate of change declines around the neighbourhood of the winning neuron. At the end of the training only the winning unit is adjusted. As a result of the self- organising process the data vectors of similar input are mapped to nearby map units in the SOM.

4. Used Data: The Two Lexical Tasks on Emotion Concepts

In our study we have used the data based on a survey that was carried out in written form during the summer months of 2003 in Estonia. The number of respondents was 100 (50 men and 50 women), aged from 14 to 76, all native speakers of the Estonian language. There were 24 emotion concepts selected for the study that form a small but representative set of the category, sharing the prototypical features of emotion concepts to various degree. The selection is based on the results of tests of free listings (Vainik 2002) as well as on word frequencies in the corpora. The participants had to complete two tasks measuring the concepts by means of different levels of knowledge.

In the first task they had to evaluate the meaning of every single word against a set of seven bipolar scales, inspired by Osgood's method of semantic differentials (Osgood, Suci, Tannenbaum 1975). The "semantic features" measured with polar scales drew qualitative (unpleasant vs. pleasant), quantitative (strong vs. weak emotion, long vs. short in duration), situational (increases vs. decreases action readiness, follows vs. precedes an event), and interpretative distinctions (felt in the mind vs. body, depends mostly on oneself vs. others). According to the theory of conceptual spaces this task addressed itself to the emotion concepts at the least abstract or subconceptual level of representations, where concepts get their value and structure via organizing the data of perceptual input according to their measurable qualities.

In the second task the same participants had to elicit emotion terms similar and opposite by meaning to the same 24 stimulus words. This task addressed itself to the emotion concepts via the most abstract or symbolic level of representations in the Gärdenfors's model. This task relayed on the speaker's intuitive knowledge about the similarities and dissimilarities of the concepts.

As the quality dimensions are claimed to be the main tool of similarity judgements, the underlying structure of conceptual space of emotion terms should not depend on the nature of the task the data are gathered in. From whatever direction to approach, the inherent and universal structure of the conceptual space should stay intact and to show up brilliantly. This was the assumption before applying the process of self-organization of SOM program[1] to our data.

5. The Self-Organizing Maps of the Emotion Concepts

Figure 1 pictures the conceptual space of emotions in Estonian according to the results of the first task, and Figure 2 according to the second task. From the very first glimpse it is clear, that the structure of concept organization approached via different levels of knowledge is not identical, though.

The SOM of the first task appears as a bilaterally symmetrical representation where the differences of judgments accumulate in the middle of the chart as a dark area. The darker is the colour on the graph, the bigger are the differences in the semantic profiles of the emotion terms. The positive emotion concepts tend to gather to the upper part of the graph and the words referring to negative emotions to the lower part of the graph.

The concepts are situated on the edges of the graph only, what means, that the similarity of the neighbouring nodes is big enough and the discrepancies from the nodes situated on the opposite side of the graph are big and systematic enough, too. The main organizing dimension of the representations appears to be the negativeness and positiveness of the concepts, that extends the shape of the SOM map in one direction. This is, however, a higher order dimension as compared to the dimensions of evaluated qualities (see Vainik 2004 for a closer discussion). As the anticipatory states (*fear, excitement, concern*), gathered to the right edge of the graph, the scale *follows* vs *precedes an event* seems to function as an additional dimension of the conceptual space. This is a situational characteristic inherent in some of the selected emotion terms.

The inter-correlations of the variables are presented in Table 1. There is a quite notable negative correlation between the scales *increases* (vs. *decreases*) *action readiness* and *unpleasant* (vs. *pleasant*). A distinction between positive and negative emotions (Figure 1) and high correlation with action readiness gives us the main idea of the functional quality dimension of giving two-valenced feedback among the emotion concepts. We conclude that the dimensions of hedonistic (*pleasantness-unpleasantness* and motivational (*increase-decrease action readiness*) evaluations appear as inseparable, integral dimensions co-occurring in the meanings of Estonian emotion terms and consisting the main structure of the conceptual space.

[1] The SOM Toolbox made by researchers at Helsinki University of Technology (HUT).

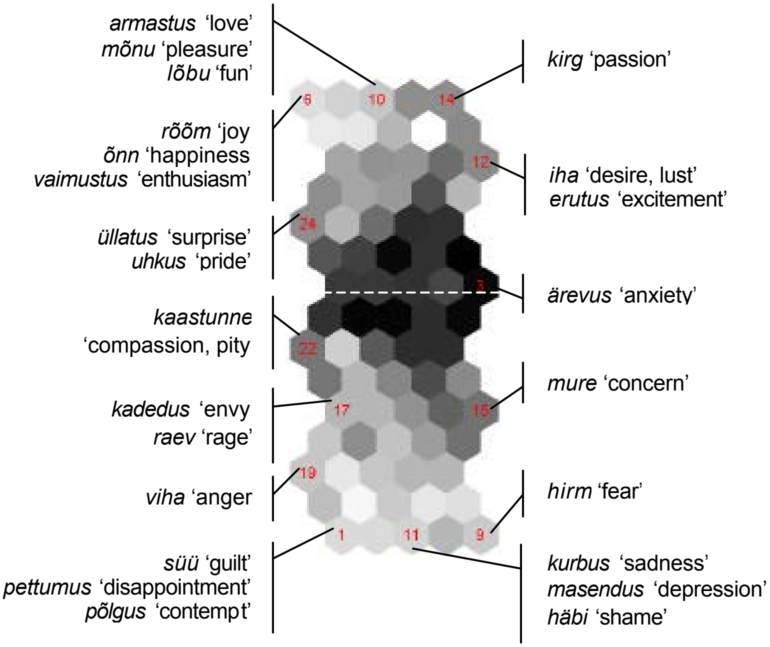


Figure 1. The locations of 24 concepts of emotion in the Estonian language on the self-organising map according to the evaluations on seven scales.

Table 1. Correlations of variables

ID	Joint scale	1.	2.	3.	4.	5.	6.	7.
1.	strong (vs. weak) emotion	—	-.041	-.028	**.253**	.032	.157	-.162
2.	follows (vs. precedes) an event		—	**.239**	-.008	-.060	-.079	.121
3.	felt in the mind (vs. body)			—	.093	.050	-.031	.122
4.	long (vs. short) in duration				—	**.137**	.034	-.045
5.	depends mostly on oneself (vs. others)					—	.002	-.017
6.	increases (vs. decreases) action readiness						—	**-.720**
7.	unpleasant (vs. pleasant)							—

The SOM of the second task (Figure 2) presents the structure of the same set 24 concepts together with the 71 most frequently elicited "similar" and "opposite" terms. Instead of bilateral symmetry and differences accumulating in the middle (Figure 1) we can see here the similarities to accumulate in the middle (the bright area) and instead of gathering to the edges, the concepts are situated throughout the whole graph. Closer look at the data reveals that the central part of the graph consists of the "opposite" concepts, missing some prototypical emotional quality (see Vainik 2004 for a closer information).

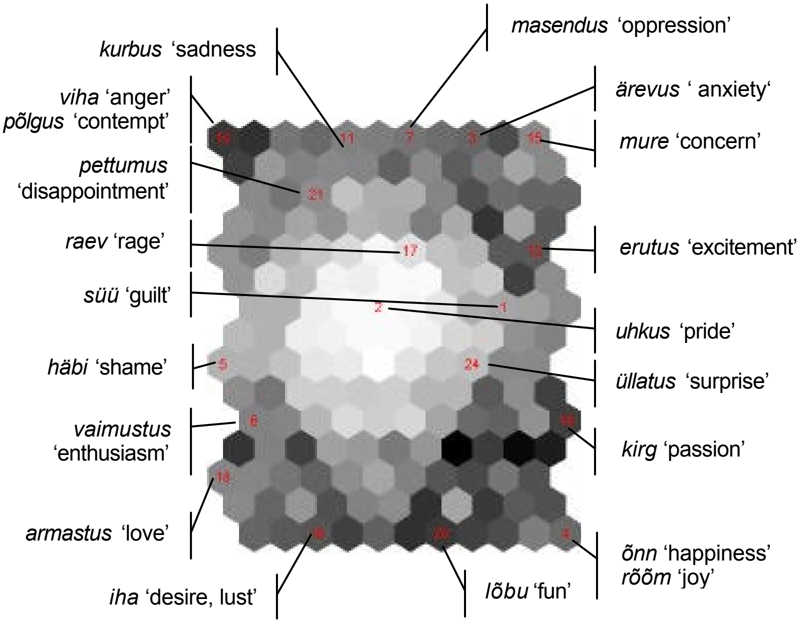


Figure 2. A self-organizing map of emotion concepts based on the relations of similarity and oppositeness.

The overall organization of the graph is radially symmetrical. There are complementarily matching (positive vs. negative) counterparts of affective states sitting in the opposite corners of the graph: positive reactional states match negative reactional states; positive proactional states match negative ones. Symmetrical are also the edges of the graph between the corners of high activation. So a positive hedonistic state matches antihedonistic states, and states of positive social feedback match the states of getting negative feedback from social interaction, all of a relatively low activation.

The most important dimension in the results of the second task seems to be the level of activation. The division of concepts into positive and negative ones is related to feedback functions and is therefore many-folded and holds for specific types and aspects of the emotional situation in which the feedback takes place.

6. A conclusion: where do the conceptual spaces come from?

The Self-organizing maps (Figures 1 and 2) as the main results of differently accessed semantics of Estonian emotion terms do not look identical. The data produced by the informants about one and the same set of stimulus words organized itself differently according to the level of abstractness the conceptual knowledge about emotions was accessed at.

The subconecptually accessed knowledge self-organized itself as a bilaterally symmetrical conceptual space with one dominating higher order dimension (positivity-negativity). The symbolically (via the relations of antonymy and synonymy) accessed knowledge self-organized itself as a radially symmetrical conceptual space, where the level emotional activation seemed to be the main hidden organizing dimension. In the second task of our model the concepts of emotions didn’t have enough information to share. It means the concepts were presented only by relations between them but there was no information in the subconceptual level. As a result we could see on the graph (Figure 2) that the specific quality dimensions, that could describe the emotions, are missing.

There seems to be no such thing as one independent conceptual space of emotions in the form of fixed network of interrelated emotion concepts determined by a fixed number of dimensions holding for most speakers of Estonians nor are the experimentally self-organized conceptual spaces independent of the nature of source data (numerical self-ratings vs. lexical production) and the level of access. All there is shared is a rather general division of the emotion terms into positive and negative ones and the flexibility to apply these concepts to ones exp eriences of positive and negative feedback in the course of intra- and interpersonal communication.

Gärdenfors (2000a: 228) noted that humans have powerful abilities to detect multiple correlations among different domains. In theory of conceptual spaces, this kind of inductive process corresponds to determining mappings between the different domains of a space. Using such mapping, one can then determine correlations between the regions of different domains.

The method of SOM was used as an independent analytical tool and as an analogy of the network model of human data processing. One should not forget, however, that any visually attractive representation of conceptual space or emotion *qualia* cannot be identified either with spatial dimensions or with distances between the nodes of a real "wet" neural network. Despite the fact that we couldn't construct an exact presentation of cognitive processes taking place in the brain at least we could get some insight into the space of concepts.

References

Aitchison, J. 2003. *Words in the mind: An introduction to the mental lexicon* (3rd ed.). Blackwell Publishing.

Deboeck G., Kohonen T. 1998. (eds.) Visual explorations in finance: with self-organizing maps, Berlin: Springer

Cruse, A. 2000. *Meaning in language. An introduction to semantics and pragmatics*. Oxford: Oxford University Press.

EKG I = Eesti keele grammatika I. [Estonian grammar I.] 1995. M. Erelt, T. Erelt, H. Saari, Ü. Viks. (Eds.).Tallinn: Eesti Teaduste Akadeemia Eesti Keele Instituut.

Gärdenfors P. 2000a. Conceptual Spaces The Geometry of Thought, The MIT Press: London

Gärdenfors P. 2000b. Concept combination: a geometrical model, pp. 129-146 in L. Cavedon, P. Blackburn,, N. Braisby and A. Shimojima (eds) *Logic language and Computation* Vol 3, CSLI, Stanford, CA.

Helsinki University of Technology. *The SOM Toolbox version 2*, Retrieved November 20, 2000, from http://www.cis.hut.fi/projects/somtoolbox/

Kohonen T. 2000. Self-organising maps (3rd edition), Berlin: Springer

Langacker, R. 1987. *Foundations of cognitive grammar I. Theoretical Prerequisites*. Stanford: Stanford University Press.

Osgood, C. E:, Suci, G.. J., Tannenbaum, P. H. 1975. The Measurement of Meaning. Urbana and Chicago: University of Illinois Press.

Russell, J. A. 1980. A circumflex model of affect. *Journal of Personality and Social Psychology* 39, 1161–1178.

Vainik, E. 2002. Emotions, emotion terms and emotion concepts in an Estonian folk model. *Trames*, 6(4), 322–341.

Vainik, E. 2004. Lexical knowledge of emotions: The structure, variability and semantics of the Estonian emotion vocabulary. Tartu: Tartu Ülikooli Kirjastus.

Viberg, A. 1994. Vocabularies. Bilingualism in deaf education. *International Studies on Sign Language and Communication of the Deaf 27*, 169– 199.

Watson, D. & Tellegen A. 1985. Toward a consensual structure of mood. *Psychological Bulletin*, 98, 219-235.

TOOMAS KIRT works for Estonian Financial Supervision Authority as IT supervisor. In 1999 he received degree M. Sc. in Information Processing at Tallinn University of Technology (Thesis: “Self Organising Maps of Estonian Banks”). Currently he is doctoral student at Tallinn University of Technology and his fields of study are data analysis, self-organization and artificial intelligence. In spring 2001 he studied one term at Helsinki University of Technology. E-mail: Toomas.Kirt@mail.ee.

ENE VAINIK is a senior researcher at the Institute of the Estonian Language. Her main point of scientific interest lays in cognitive semantics and psycholinguistics. She has introduced the Ronald Langacker’s Cognitive Grammar the first in Estonian (a case study of the Estonian external locative cases) and investigated the emotion vocabulary of emotions. In 2003 the Phd thesis “Lexical knowledge of emotions. The structure, variability and semantics of the Estonian emotion terms “ was published. Ene.Vainik@eki.ee

APPLICATION OF DIALECTAL AUDIO DATA IN CREATING LANGUAGE RECOGNITION PROGRAMMES

Asta Leskauskaitė, Daiva Vaišnienė
Institute of Lithuanian Language, Vilnius, Lithuania

Abstract

In the spoken language one and the same segment can be of various lengths and its internal structure can vary greatly, and that leads to serious flaws in applying recognition programmes. To eliminate these shortcomings the application of the principle of phonetic recognition is proposed. In creating programmes for the Lithuanian language account should be taken not only of the standard language but also of the dialects, more precisely of the dialectal phonetic sound variants and their acoustic peculiarities. At the present time several sources, containing records of Lithuanian dialects, are available for research.

Keywords: speech corpora, phonetics, dialects, acoustic features, variant

1. Introduction

The issues of language recognition have been investigated for over three decades in many places all over the world. The sphere of the technological application of linguistic signals is most varied: aids for the disabled, criminal investigation and defence, management and preservation of cultural heritage, administration, telecommunications, etc. However, despite the fact that in creating language recognition programmes recourse is made to techniques taking into account the variance of speech segments, the achieved results are not always satisfactory (Rudžionis 1998; BTTLKAPVKP 2001; Lipeika, Lipeikienė, Telksnys 2002).

2. The principle of phonetic recognition

These shortcomings could be eliminated by applying the principle of phonetic recognition and creating speech corpora. In order to create a universal corpus, it is necessary to have a comprehensive description of the diversity of the phonetic segments of a particular language, to determine the effect of the context and the stylistic and idiosyncratic varieties.

At present, in establishing new databases of linguistic signals, use is made of the ready speech corpora accumulating various materials (TIMIT, NTMIC, CTIMIC, ISOLET, OGI etc.). The TIMIC speech corpus is used probably most frequently, since the boundaries of the phonemes collected there are marked very thoroughly. The LTDIGITS, created jointly by the University of Vilnius and the Technological

University of Kaunas and now undergoing tests, is an attempt to produce the first Lithuanian speech corpus (BTTLKAPVKP 2001).

3. Speech variants

As has already been mentioned, one of the spheres of the practical application of speech recognition technologies is the handling and preservation of cultural values, including the processing of linguistic data and their effective exploitation for research purposes. In this respect, in creating language recognition programmes, some factors should be taken into consideration.

The standard language Lithuanian language recognition programmes are usually based on the data of the standard language, which, however, is a kind of artificial phenomenon rather than a naturally derived language (see Figure 1). It is a sort of a gauge but not a reflex of a real linguistic situation. Consequently, its data are insufficient for the presentation of the full diversity of possible speech sounds.

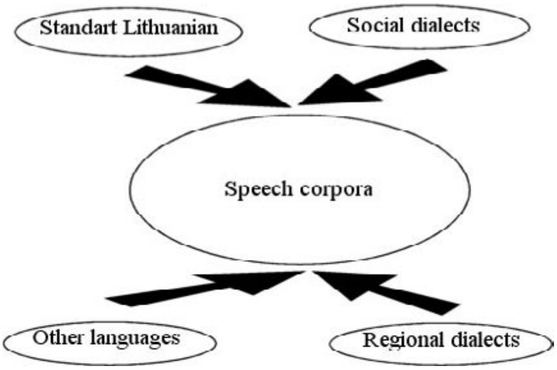


Figure 1. Speech variants

Social dialects In the natural speech of many individuals we can hear a number of phonetic, lexical and other elements alien to the standard language. They are linked with certain factors, such as the speakers' birthplace (village, town or city), education, age, etc. The inhabitants of towns tend to speak the standard language; nevertheless, dialectal traces can be detected in their speech, and it is often possible to distinguish a speaker of one dialect from that of another by ear. Therefore sociolinguistic variations cannot be ignored in creating speech corpora and language recognition programmes.

Other languages The third factor which should be taken into account is other languages, spoken in the area. In Lithuania the speech habits of the people, whose native tongue and the acquired accent are other than Lithuanian, influence the way they speak Lithuanian. The acoustic and articulatory peculiarities of their speech, if ignored, can impair the linguistic recognition process.

Regional dialects No less important is the incorporation of dialectal phonetic variants into the linguistic databases. This factor was emphasized by the US researchers in their creation of the widely known TIMIC database, now considered a standard in this sphere. Linguistic databases, devoid of the dialect materials, would be of limited application possibilities.

4. Sources

In Lithuania as in other countries the creation of modern language recognition programmes is based on the phonetic speech recognition. The essence of this technique is the determination of the boundaries of separate sounds in the word and the description of their acoustic and articulatory features, taking into consideration co-articulatory factors. The results of linguistic research and dialect recordings could be used in preparing systems, which could properly discriminate speech elements (see Figure 2).

In Lithuania experimental techniques have been employed in the phonetic investigation of the standard language and its dialects for several decades (EPFKM 1968, 1972, 1974; Girdenis 1995; Atkočaitytė 2002; Leskauskaitė 2004 et al.). Variously recorded dialectal data were selected and segmented. Subsequently sonograms and oscillograms were analyzed, the boundaries of the formant frequencies of sounds and the variance intervals of separate acoustic features, etc. were determined.

Doubtless, account was also taken of the effect of intonation and adjacent sounds on the changes of the sounds under investigation. Thus, in some Lithuanian subdialects the quality of the short vowel *i* greatly depends on the neighbouring sounds. Preceded by a velar consonant *l*, it can be similar to the Polish sound *y* or Russian *ы*, and after *r, š, ž* it is more fronted, but different than the one after *m* or *k*. This vowel can be affected by the closeness or openness of the adjoining syllable, etc. Nevertheless, despite the difference of the acoustic and articulatory features of the variants, they all mark the same vowel. Various research techniques define the limits of the variation of the sound *i*, i. e. they establish the lowest and highest indexes of its acoustic peculiarities and intensity.

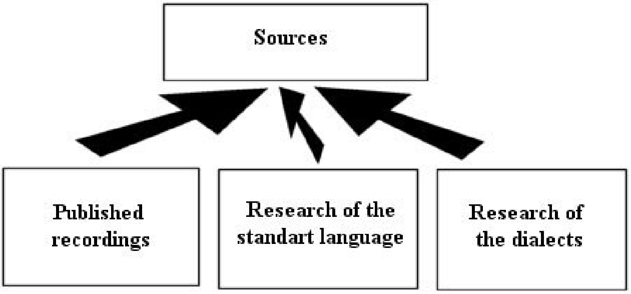
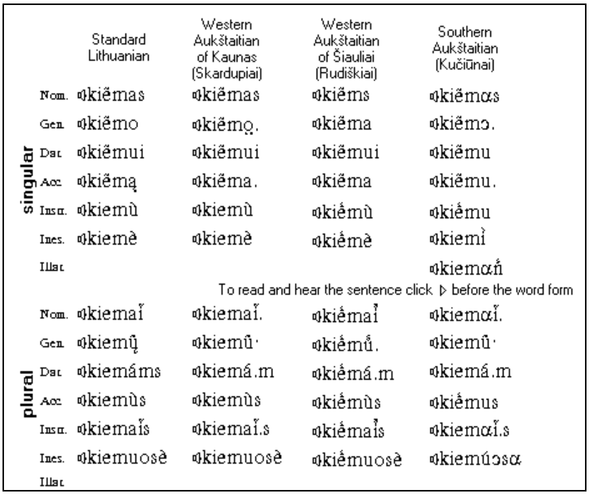


Figure 2. The Sources

The same is true of other sounds. A speech recognition programme should either reject or identify all variants as the vowel *i* (that would depend on the purpose of the expected results).

At present several sources, containing records of Lithuanian dialects, such as the CDs *Lietuvių tarmės. I dalis / Lithuanian Dialects. Volume I* (2002) (and its Internet version) and *Lietuvių kalbos tarmių chrestomatija* [Reader of Lithuanian Dialects] (2004). are accessible to the researchers.

These sources and the second part of the multimedia dialect dictionary (now under preparation) contain various dialectal recordings (words, phrases and sentences). The presentation of entire paradigms enables their direct comparison, and that shows that the specimens of the standard language differ from those of the dialects both in their morphological and phonetic expression (see Figure 3).



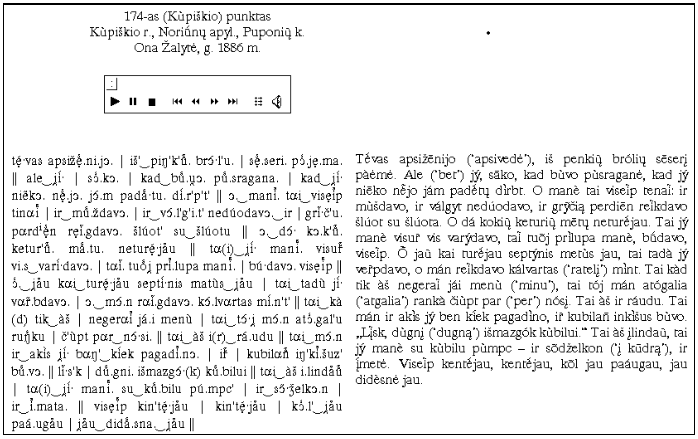
		Standard Lithuanian	Western Aukštaitian of Kaunas (Skardupiai)	Western Aukštaitian of Šiauliai (Rudiškiai)	Southern Aukštaitian (Kučiūnai)
singular	Nom.	🔈kiẽmas	🔈kiẽmas	🔈kiẽms	🔈kiẽmɑs
	Gen.	🔈kiẽmo	🔈kiẽmọ.	🔈kiẽma	🔈kiẽmɔ.
	Dat.	🔈kiẽmui	🔈kiẽmui	🔈kiẽmui	🔈kiẽmu
	Acc.	🔈kiẽmą	🔈kiẽma.	🔈kiẽma	🔈kiẽmu.
	Inst.	🔈kiemù	🔈kiemù	🔈kiế̄mù	🔈kiế̄mu
	Ines.	🔈kiemè	🔈kiemè	🔈kiế̄mè	🔈kiemi̓
	Illat.				🔈kiemɑń
			To read and hear the sentence click ▷ before the word form		
plural	Nom.	🔈kiemaĩ	🔈kiemaĩ.	🔈kiế̄maĩ	🔈kiemɑĩ.
	Gen.	🔈kiemų̃	🔈kiemũ·	🔈kiế̄mũ.	🔈kiemũ·
	Dat.	🔈kiemáms	🔈kiemá.m	🔈kiế̄má.m	🔈kiemá.m
	Acc.	🔈kiemùs	🔈kiemùs	🔈kiế̄mùs	🔈kiế̄mus
	Inst.	🔈kiemaĩs	🔈kiemaĩ.s	🔈kiế̄maĩs	🔈kiemɑĩ.s
	Ines.	🔈kiemuosè	🔈kiemuosè	🔈kiế̄muosè	🔈kiemúɔsɑ
	Illat.				

Figure 3. Declencion of the word *kiemas* (farmyard)

Dialect data are presented in greater detail and more systematically in *Lietuvių kalbos tarmių chrestomatija*. This publication acquaints the user with all Lithuanian dialects and their phonetic, morphological and other peculiarities. The words and larger segments are presented in phonetic transcription, to some extent reflecting the pronunciation of each sound. The majority of the discussed peculiarities is illustrated by audio recordings. Besides, each dialect word is accompanied by its equivalent in the standard language, and that makes their contrast more marked.

It is worth noting that sound differences are presented in the *Reader* hierarchically, that is, the main distinguishing feature is followed by minor traits, peculiar to subdialects and this type of characterization is carried on up to the peculiarities of the smallest territorial dialect varieties. Doubtless, the dialectal identification of a particular speaker could be made on the basis of the main data, representative of the dialect.

A great deal of information could be derived from the fieldwork recordings (see Figure 4). True, not all the records are of desirable quality, but at least a part of them are quite analyzable. It is important that the stories, as specimens of natural speech, are rendered in a normal ordinary intonation, which has much to do with acoustic and articulatory features of sounds. This factor is crucial for the recognition of everyday speech cases.



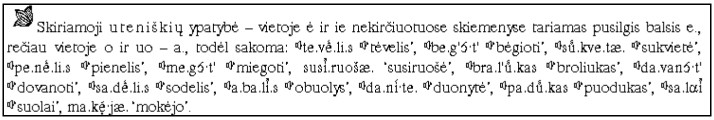
174-as (Kùpiškio) punktas
Kùpiškio r., Noriū́nų apyl., Puponių̃ k.
Ona Žalytė, g. 1886 m.

tẹ̄·vas apsižẹ̄.ni.jɔ. | iš'‿piŋ'k'ů. brɔ́·l'u. | sẹ̄.seri. pɔ̊́.jẹ.ma. || ale‿jí̯· | sɔ̊́.kɔ. | kad‿bů̊.u̯ɔ. pů̊.sragana. | kad‿jí̯· niẽkɔ. nẹ̊́.jɔ. jɔ́.m padå̊·tu. dí.r'p't' || ɔ.‿manĩ. tαi‿visẹĩp tinαĩ | ir‿mů̊.ždavɔ. | ir‿vɔ́.l'g'i.t' nedúodavɔ.‿ir | grĩ·č'u. pαrd'ẹ̃n rẹĩ.gdavɔ. šlúot' su‿šlúotu || ɔ.‿dɔ́· kɔ.k'ů̊. ketur'ů̊. må̊.tu. neturẹ́·jåu || tα(i)‿jí̯· manĩ. visuȓ vi.s‿varí·davɔ. | tαĩ. tuɔ̊̃j̯ prĩ.lupa manĩ. | bú·davɔ. visẹĩp || ɔ̊́.‿jåu kαi‿turẹ́·jåu septí·nis matùs‿jåu | tαi‿tadù jí̯· vαȓ.bdavɔ. | ɔ.‿mɔ́.n rαĩ.gdavɔ. kɔ́.lvαrtas mí.n't' || tαi‿kà (d) tik‿àš | negerαĩ já.i menù | tαi‿tɔ́·j̯ mɔ́.n atɔ̊́.gal'u ruŋ̃ku | č'ùpt pαr‿nɔ́·si. || tαi‿àš i(r)‿rá.udu || tαi‿mɔ́.n ir‿akĩs jí̯· bαŋ'‿kíek pagadĩ.nɔ. | iȓ | kubilαñ iŋ'kĩ.šuz' bů̊.vɔ. || lĩ·s'k | dů̊.gni. išmazgɔ́·(k) ků̊.bilui || tαi‿àš i.lindåů̊ | tα(i)‿jí̯· manĩ. su‿ků̊.bilu pú.mpc' | ir‿sɔ̃·ǯelkɔ.n | ir‿ĩ.mata. || visẹĩp kin'tẹ́·jåu | kin'tẹ́·jåu | kɔ̊́.l'‿jåu paá.ugåu | jåu‿didå̊.sna.‿jåu ||

Tė́vas apsižēnijo ('apsivedė'), iš penkių̃ brólių sēserį pàėmė. Ale ('bet') jý, sãko, kad bùvo pùsraganė, kad jý niẽko nẽjo jám padė́tų dìrbt. O manè tai viseĩp tenaĩ: ir mùšdavo, ir válgyt nedúodavo, ir grỹčią perdiẽn reĩkdavo šlúot su šlúota. O dá kokių̃ keturių̃ mẽtų neturė́jau. Tai jý manè visuȓ vis varýdavo, taĩ tuõj prĩlupa manè, bū́davo, viseĩp. Õ jaũ kai turė́jau septýnis metùs jau, tai tadà jý veȓpdavo, o mán reĩkdavo kálvartas ('ratelį') mìnt. Tai kàd tik àš negeraĩ jái menù ('minu'), tai tój mán atógalia ('atgalia') rankà čiùpt par ('per') nósį. Tai àš ir ráudu. Tai mán ir akìs jý ben kíek pagadìno, iȓ kubilañ inkìšus bùvo. „Lį̃sk, dùgnį ('dugną') išmazgók kùbilui." Tai àš įlindaũ, tai jý manè su kùbilu pùmpc – ir sõdželkon ('į kūdrą'), ir įmetė. Viseĩp kentė́jau, kentė́jau, kõl jau paáugau, jau didèsnė jau.

Figure 4. Distinguishing Features of the Western Aukštaitian Subdialect of Šiauliai

Once again it is worth noting that both the *Dictionary* and the *Reader* indicate the principal distinguishing features (see Figure 5). This speech corpus and the achieved results of the phonetic studies of nearly all Lithuanian subdialects are a solid groundwork for the enhancement of a speech corpus of the standard language and a comprehensive description of the peculiarities of an authentic spoken language.



Skiriamoji uteniškių ypatybė – vietoje ė ir ie nekirčiuotuose skiemenyse tariamas pusilgis balsis e., rečiau vietoje o ir uo – a., todėl sakoma: 🔈te.vẽ.li.s 🔈'tėvelis', 🔈be.g'ɔ́·t' 🔈'bėgioti', 🔈sů̊.kve.tæ. 🔈'sukvietė', 🔈pe.nẽ.li.s 🔈'pienelis', 🔈me.gɔ́·t' 🔈'miegoti', susĩ.ruošæ. 'susiruošė', 🔈bra.l'ů̊.kas 🔈'broliukas', 🔈da.vanɔ́·t' 🔈'dovanoti', 🔈sa.dẽ.li.s 🔈'sodelis', 🔈a.ba.lĩ.s 🔈'obuolys', 🔈da.ní·te. 🔈'duonytė', 🔈pa.dů̊.kas 🔈'puodukas', 🔈sa.lαĩ 🔈'suolai', ma.kẹ́·jæ. 'mokėjo'.

Figure 5. Text of the Eastern Aukštaitian Subdialect of Kupiškis

Additionally, the dialect speech recognition programmes are highly necessary for dialect and socio-linguistic investigations based on the spoken materials. One of the challenges facing linguists and informatics specialists is to speed up the research process to enable the change of the sound into a graphic record.

5. Conclusions

In creating speech recognition programmes and databases the aforementioned factors should be taken into account: the obtained results of dialect investigations and sound recordings should be put to use and lists of words stored in databases should be enlarged. New facts would hone speech recognition programmes and increase the trustworthiness of their results. In future such programmes could be adapted to everyday needs and the sphere of their usage be widened. One of such applications could also be an automatic recognition and facilitated processing of dialect data in scholarly activities.

References

Atkočaitytė, Daiva 2002. *Pietų žemaičių raseiniškių prozodija ir vokalizmas*. Vilnius: Lietuvių kalbos instituto leidykla.

Girdenis, Aleksas 1995. Lietuvių kalbos bei jos tarmių prozodinių reiškinių ir fonemų alofonų analizė. Mokslinė ataskaita. Vilnius: Vilniaus universitetas.

EPFKM 1968. Eksperimentinės fonetikos ir kalbos psichologijos kolokviumo medžiaga 3. Vilnius: Vilniaus valstybinis pedagoginis institutas.

EPFKM 1972. *Eksperimentinės fonetikos ir kalbos psichologijos kolokviumo medžiaga* 5. Vilnius: Vilniaus valstybinis pedagoginis institutas.

EPFKM 1974. *Eksperimentinės fonetikos ir kalbos psichologijos kolokviumo medžiaga* 6. Vilnius: Vilniaus valstybinis pedagoginis institutas.

Lipeika, Antanas; Lipeikienė, Joana; Telksnys, Laimutis 2002. Development of isolated word speech recognition system. In: *Informatica* 13 (1). 37–46.

Lietuvių kalbos tarmių chrestomatija. Vilnius: Lietuvių kalbos instituto leidykla, 2004.

Lietuvių tarmės. I dalis / Lithuanian dialects. Volume I. Vilnius: Lietuvių kalbos instituto leidykla, 2001.

Rudžionis, Vytautas 1998. Priebalsių atpažinimas difonuose priebalsis – balsis. *Informacijos mokslai* 9. 47–54.

BTTLKAPVKP 2001. Balso technologijų taikymo lietuvių kalbai analizė ir perspektyvinių veiklos krypčių pagrindimas, Kaunas, Vilnius. From http://www.likit.lt/all/balso_tech/01_ivadas.htm#up.

Leskauskaitė, Asta 2004. *Pietų aukštaičių vokalizmo ir prozodijos bruožai*. Vilnius: Lietuvių kalbos instituto leidykla.

ASTA LESKAUSKAITĖ is a researcher at the Institute of Lithuanian Language's Department of Language History and Dialectology. Qualification: PhD (2001) (thesis title "The prosody and vocalism of the Southern Aukštaitian dialect"). Her research interests concern dialectology, phonology, experimental fonetics, morphology, etc. Leskauskaitė has worked on a number of projects, includings several Lithuanian language CDs and creating Dialect database. She also the author or co-author of numerous scholarly publications. E-mail: astal@ktl.mii.lt

DAIVA VAIŠNIENĖ is a researcher at the Institute of Lithuanian Language's Department of Language History and Dialectology. Date of birth Dr. philol. Daiva Vaišnienė: 1972 05 28. Place of birth: Lithuania. In 1995 graduated from Vilnius pedagogical university (master of Arts). Qualification: PhD (2000) (thesis title "The prosody and vocalism of the Southern Žemaitian dialect of Raseiniai"). Present employer: Institute of Lithuanian language. Position held: research secretary, Head of Dialect Archive. Main activities: creating of Multimedia dictionary "Lithuanian dialects", organizing of dialectal sound records and wrotten texts, creating Dialects database. E-mail: atko@ktl.mii.lt.

LEXICOGRAMMATICAL PATTERNS OF LITHUANIAN PHRASES

Rūta Marcinkevičienė, Gintarė Grigonytė
Vytautas Magnus University, Kaunas, Lithuania

Abstract

The paper overviews the process of compilation of the first corpus-based Dictionary of Lithuanian Phrases. Phrases are transformed from collocational strings which were extracted from the corpus of contemporary Lithuanian language of 100 million running words applying a new statistical method called Gravity counts. The paper presents theoretical approach towards the most relevant notions of collocation, collocational string, phrase, part of speech, grammatical and lexicogrammatical pattern. Statistical method of extraction of collocational strings is shortly presented together with the initial output of raw collocational strings. Types of transformations of collocational strings into phrases and other manual procedures are described in a nutshell while primary results of patterning of the Lithuanian phrases as well as future steps are presented in greater detail.

Keywords: collocation, collocational string, Gravity counts, fragment of text, POS pattern, grammatical pattern

1. Introduction

The compilation of the Dictionary of Lithuanian Phrases includes three main phases: extraction of collocational strings from the corpus of present day Lithuanian language, transformation of collocational strings into phrases, and patterning of all the phrases. Each phase is based on certain theoretical approaches as well as notions of collocation, collocational string, phrase, and pattern, presented here.

Collocation is a fuzzy term embracing a great variety of notions. The definition of a collocation differs according to researcher's standpoint and the method of extraction. There are two different perspectives on the notion of collocation from the point of view of its form and structure. One group of authors (J.Firth, J.Sinclair, M.Stubbs, among others) prefers contextual or statistical definition of collocation. It could be generelised as follows: one item collocates with another that appears somewhere near it in a given text. The assumption underlying collocation is based on its structure: collocation consists of a node word and its collocates, so the search of a collocation starts with the node word. Thus statistical definition highlights lexical relationship between two or more items that tend to co-occur. However, it does not allow one to detect multi-word collocations as they appear in the texts and to define their boundaries. Statistical collocations are usually presented as lemmas for node words and their collocates.

Another group of authors (G.Kjellmer, G. Williams, etc.) pursue a lexicographic approach and include grammatical well-formedness and grammatical acceptability in the list of criteria for collocations. They see collocation as a fragment of text, not as a list of collocates for the previously selected node words.

A compiler of a dictionary of collocations has to choose between the two approaches, since the attitude towards collocation predetermines the method of extraction and the method of presentation. From the perspective of the Lithuanian language the lexicographic approach is more acceptable. It provides a lexicographer with authentic strings of words or fragments of texts obtained by applying statistical tools. These strings contain collocating grammatical forms presented in their natural word order (and not isolated lemmas) which are of paramount importance for the highly inflected Lithuanian language. Such *collocational strings* can be sorted out with their grammatical autonomy in mind but they do not have to be reconstructed from a mere list of nodes and their collocates.

Finally, this approach allows us to avoid making a pre-selected list of node words and to process the entire corpus from the first to the final word. It presents, therefore, a full-text approach to language and utilises the entire corpus, i.e. every sentence it contains, not merely concordances derived from the corpus on the basis of a previously compiled list of node words. Consequently this approach allows us to determine the amount of text that is formed on the idiom principle (Sinclair 1991: 109-121). The choice of the lexicographic approach as opposed to the statistical one determines the choice of a particular method for the extraction of collocations.

Collocational strings after they are extracted from raw texts do not always coincide with grammatically well formed and semantically sufficient word combinations, therefore they have to be transformed into such autonomous phrases either by cutting off irrelevant or adding their missing parts. By *phrase* we understand as a frequently used autonomous fragment of text. Our theoretical standpoint does not allow us to interfere with the inner structure of a phrase and to change its word order or morphological form.

The last phase in the compilation of the dictionary is patterning of phrases. The concept of a pattern is borrowed from corpus linguistics where it is conceived as a juncture of most prominent lexical and grammatical features of a phrase (Hunston et al. 2000). Traditionally patterns are centered either around a lexical item (in lexicography) so that they reveal its usage and meaning or they are centered around a part of speech (e.g. verbal, nominal, adjectival patterns in grammar). In the first case patterns are too concrete and specific, in the second case they are two abstract and general.

We apply a holistic approach and aim at combining of a) lexical, b) semantic and c) grammatical features into one pattern, e.g.:

verb of motion (b) + preposition "link" (a) + concrete noun in Genitive (b-c).

Our basis of patterning is different from the existing corpus or traditional approaches due to the source material. It is based on the list of phrases of various lengths instead of invented examples or node word concordances. Since phrases and collocations represent the most frequent and significant fragments of the Lithuanian language, patterns derived from these phrases can be regarded as basic. As such they can be of paramount importance for the probabilistic parser and other related tools.

2. Method of extraction of collocational strings

Collocational strings were extracted from the corpus of Lithuanian language with the help of a statistical method called Gravity counts. It adopts a linear approach of consecutive counts of words in a text, and of all the texts in a corpus, based as it is on the combinability counts of each pair of words in the corpus irrespective of their hierarchical status, i.e. there is no *a priori* list of node words for which collocates are obtained from the corpus. Each word in the corpus is processed as the node word; its gravity by reference to the pairing word and the next two words in the span of three words is calculated using the formula below (for more about the method see Daudaravičius et al. 2004):

$$G(x,y) = \log\left(\frac{f(x,y)\cdot n(x)}{f(x)}\right) + \log\left(\frac{f(x,y)\cdot n'(y)}{f(y)}\right)$$

Gravity Counts are based on an evaluation of the combinability of two words in a text that takes into account a variety of frequency features, such as individual frequencies of words, the frequency of a pair of words and the number of different words in the selected span. Gravity Counts highlight habitual co-occurrence of two words in a text within the chosen span, in our case the span of three words. If the first word *x* is used more habitually than expected in front of the second word *y*, and the second word *y* is used more habitually than expected after the first word *x*, then *x* and *y* form a minimal collocational string.

Gravity Counts are also based on word order, so that for each first word *x* in a pair the frequency of the following three words is taken into consideration, while for each second word *y* of a pair the frequency of the three preceding words is computed. Therefore *n(x)* is the number of different words to the right of *x* and *n'(x)* words to the left of *y*; *f(x)* and *f(y)* is the frequency of *x* and *y* in the corpus.

The method of Gravity Counts and the detection of collocation boundaries helps to identify segments of texts as statistically significant colocational strings. These strings can be said to be always natural since they present authentic fragments of a text. Nevertheless, statistical collocational strings differ from the point of view of their grammatical and lexical autonomy, which is the most relevant feature in our analysis. Certain collocational strings are self-sufficient and can be regarded as autonomous and grammatically well-formed phrases. Other kinds of collocational strings are somewhat deficient and have to be transformed into a phrase. In order to differentiate between autonomous and deficient collocational strings obtained from the corpus using the method of Gravity counts, as well as to define their ratio, a manual analysis is performed. A fairly high percentage, i.e. 82 % of collocational strings are found to be autonomous and clear-cut phrases.

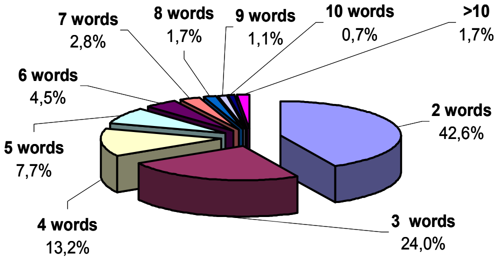


Figure 1. Distribution of initial collocational strings by their length

3. The output and its transformations

Application of Gravity Counts for the corpus of Lithuanian language resulted in processing of 110 935 000 pairs of words in the corpus of 100 million running words (1,7 million different word forms). Some pairs of words were joined into multi-word collocational strings, thus the initial list of collocational strings consists of 19,878,281 items. The list of different collocational strings is of 10,147,250 items. All the collocations cover 68.1 % of the corpus.

The output of the calculations is the list of collocational strings of varying length. The general tendency for the length of collocational strings is the same as for the frequency of words, i.e. the longer strings are less frequent than the shorter ones. The majority of strings (8,462,626 items which form 42 % of all the list) are made up of two words. The list of three-word strings is twice as short (4,760,991 items, 24 % of the list), the same can be said of the four word strings (2,629,953 items, 13 % of the list) and the five word strings (1,532,370 items, 8 % of the list). The decrease in number for the longer strings is somewhat less (see Figure 1). A typical long collocational string is taken from governmental decrees and consists of 34 words (for a more detailed description see Marcinkevičienė 2004).

The manual processing of the raw output, i.e. transformation of statistical collocational strings into well-formed phrases, consists of several steps and procedures. The first step is to delete all rare strings, irrespective of their length (1 to 3 occurrences) and some more frequent strings depending on their length: two-word strings up to 19 occurrences, three word strings up to 9 occurrences, four word strings up to 8 occurrences, five word strings up to 4 occurrences. This arbitrary decision was based on the considerable amount of noise in these particular items. Besides, only these collocational strings were left that contained at least one noun. This was an arbitrary decision cause by manually unmanagable length of the initial output, i.e. ca 20 million items.

The remaining or intermediate list of 88 562 collocational strings of different length was processed applying three different procedures: lexically well-formed and grammatically autonomous collocational strings were included without changes. Some anomalous, structurally and semantically insufficient strings were deleted (e.g. parts of the string belonging to a different clause, or strings containing proper names, numbers, misprints, consisting exclusively of a noun plus conjunction, a pronoun or one of the forms of the verb “to be”) while some of them were changed. The changes include: a) shortening of grammatically irrelevant parts of long collocations, b) addition of missing words from concordances to deficient strings, mostly two or three word combinations consisting of nouns and prepositions.

The first stage of transformation, i.e. the deletion of deficient strings, left the compilers of the dictionary with the final list of 68,600. i.e. ca 74 % of the intermediate list. Addition of missing words to the deficient collocational strings (mostly consisting of prepositional phrases) did not affect the length of the list, only the length of the specific strings, e.g. some two-word dtring were transformed into three or four-word strings. Circa 10% of collocations were lengthened by adding 2-4 words to both sides of a collocation. In some cases additions consisted not of words but of parenthesis which demonstrated the gap to be filled in by a specific lexical item, e.g. a numeral, proper noun, etc.

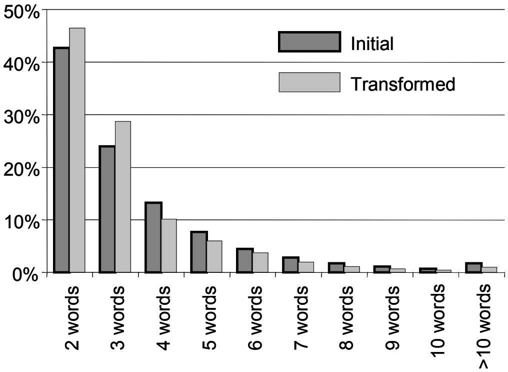


Figure 2. Distribution of initial and transformed collocational strings by their length (in number of words)

Figure 2 depicts how the length of

collocational strings was affected by the process of transformations. Two-word and three-word strings were lengthened, four and five-word strings were shortened, six-word and longer strings remained almost intact.

4. Patterns of phrases

Manually processed list of phrases revealed different nature depending on the length of a phrase. The whole list consists of three groups: phrases with more than 6 words could be called *fragments of text* due to their length, two-word phrases, on the contrary, are typical word combinations or traditional *collocations*, while the middle group, i.e. from six to three words, resembles traditional *phrases*. Four-word phrases were chosen to be patterned and described first of all here as the most intuitively recognisable and typical items, least affected by manual transformations, especially additions. The number of different items in this list is 3895, the overall frequency of all four-word phrases in the corpus is equal to 62854.

Four-word phrases were morphologically annotated using the system for morphological annotation of the Lithuanian language “Lemuoklis”. The outcome is a list of 1511 POS patterns. Majority of these patterns contained several interpretations for the same word therefore a manual disambiguation was necessary. The first step of disambiguation was to classify all four-word phrases into four groups on the bases on the number of nouns in a phrase. The result is the number of ambiguous patterns for each structural group: 152 four-noun POS patterns, 391 three-noun POS patterns, 576 two-noun POS patterns and 393 one-noun POS patterns.

Disambiguation and clustering of patterns was carried out inside the groups. The outcome of this process revealed an obvious correlation between the number of nouns in the patterns and the number of POS patterns. Four-noun phrases were identical from the point of view of their morphological structure since all the phrases consisted of nouns exclusively. Three-noun phrases presented ca 40 POS patterns (a more exact number of patterns is still the topic of ongoing discussions), POS patterns for two-noun phrases were three times more numerous, i.e. 120 POS patterns, while one-noun phrases manifested the biggest variety of possible combinations of different parts of speech – ca 300 POS patterns. POS pattern of a one-noun phrase is exemplified below:

(1) Smulkios ir vidutinės įmonės (small and medium enterprices)
(adj) (conj) (adj) (noun)

Further steps in the process of patterning of Lithuanian phrases include detailed analyses of morphological forms of different parts of speech as they are presented in the authentic non-lemmatised phrases. Additional morphological characteristics will give a more finely grained and therefore more numerous lists of grammatical patterns, e.g.:

(2) Smulkios ir vidutinės įmonės
(adj pl nominative) (conj „ir”) (adj pl nominative) (noun pl nominative)

Last but not least, grammatical patterns will be enriched with specific lexical items for auxiliary parts of speech (e.g. prepositions, particles, conjunctions) and semantic features for the groups of notional parts of speech (e.g. nouns and verbs):

(3) Smulkios ir vidutinės įmonės
(adj pl nominative) (conj „ir”) (adj pl nominative) (concrete noun pl nominative)

5. Conlusions

Extraction of collocational strings from 100 million word corpus of the Lithuanian language and their transformation into phrases revealed three main findings: a) almost 70 % of all the corpus consists of collocational strings and is based on idiom principle, b) the method of extraction is suitable since relatively small number of collocational strings were found deficient and had to be transformed into phrases manually, c) the number of POS patterns and their structural variety is big, esp. in those phrases that contain less nouns.

Acknowledgements

We are gratefull to the State Commission of the Lithuanian Language for supporting the compilation of the Dictionary of Lithuanian Phrases in 2004-2005.

References

Daudaravičius, Vidas; Marcinkevičienė, R. 2004. Gravity counts for the boundaries of collocations. In: In: Teubert, W.; Johansson, Stig (eds.) *International Journal of Corpus Linguistics* 9/2. 321-348.

Hunston, Susan; Francis, G. 2000. Pattern grammar. A corpus-driven approach to the lexical grammar of English. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Marcinkevičienė, Rūta. 2004. Dictionary of Lithuanian phrases. In: Williams, G.; Vessier, S. (eds.) *Proceedings of the Eleventh EURALEX International Congress*, *EURALEX 2004*, Lorient: UBS.741-751.

Sinclair, John. 1991. Corpus, concordance, collocation. Oxford: Oxford University Press.

RŪTA MARCINKEVIČIENĖ is head of the Department of Lithuanian language and the Centre of Computational Linguistics, Vytautas Magnus University, Kaunas. She received her doctor degree (Lithuanian language) at the University of Vilnius, dealing with comparative semantics of English and Lithuanian verbs and her degree of habilitated doctor at Vytautas Magnus University, dealing with Lithuanian corpus linguistics. Her research interests concern corpus linguistics and corpus-based lexicography, lexical semantics, pragmatics, and text linguistics (generic approach). As a visiting lecturer, she has taught Lithuanian language at the University of Stockholm, Lithuanian culture and the Theory of Genre at the University of Illinois, Chicago. She is the member of the board of the Nordic School of Language technologies. E-mail: ruta@hmf.vdu.lt.

GINTARĖ GRIGONYTĖ is an engineer-programmer of the Centre of Computational Linguistics at Vytautas Magnus University. She is doing her master studies at Kaunas Technology University, Software engineering programme. Her research interests concern computational linguistics, software engineering, automatic syntactical analysis. E-mail g.grigonyte@hmf.vdu.lt.

MODELLING PAUSES AND BOUNDARY LENGTHENINGS IN SYNTHETIC SPEECH

Meelis Mihkla
Institute of the Estonian Language, Tallinn, Estonia

Abstract

In order to make synthetic speech, generated from a given text, sound natural, one has to exert strict control over the temporal structure of the speech flow. At that, close attention should be paid to pauses and boundary lengthenings as phrase markers and vital factors of speech rhythm in general. The present study analyses the duration of pauses and boundary lengthenings in various Estonian texts (fiction, news, other) read out by 27 dictors. If we wish to lend synthetic speech a natural rhythm, it does not suffice if we just find out the mean durations of pauses and lengthenings. Instead, we should rather model their durations and temporal positions in a context-sensitive way. In this study the duration of pauses and boundary lengthenings as well as their temporal positions in synthesised speech have been modelled on the basis of text structure, using some basic methods of prediction (regression analysis). [1]

Keywords: pause, boundary lengthening, synthetic speech, regression analysis

1. Introduction

For the artificial speech to sound realistic in human ear, it should comprise natural-sounding intonation, rhythm and stress pattern. The temporal phenomena like pauses and syllable lengthenings constitute a vital part of prosodic aspects of speech. The pre-boundary lengthenings of intonation phrases are often applied in the synthesising devices, however pause modelling has less frequent currency. The reason might be that the pauses are largely variable both in duration and in location in speech flow. The duration of pauses and boundary lengthenings and their situation in speech flow depends on sentence structure, speaker and also the given language (Zvonik, Cummins 2002).

The pauses and prepausal lengthenings in the Estonian language speech have been studied cursorily or intermittently, as the by-product in the context of other tasks. Ilse Lehiste verified (Lehiste 1981) whether prepausal lengthenings were in correlation with the length of subsequent pauses and she established an extremely weak link. Diana Krull studied prepausal lengthenings in dialogue speech in two-syllable words in the

[1] This work was completed within the framework of Grant no. 5039 of Estonian Science Foundation and the state program „Estonian language and national memory“.

context of quantity degree (Krull 1997). Arvo Eek and Einar Meister looked into the end-of-sentence lengthenings on the basis of tempocorpus (Eek, Meister 2003). However, they held under scrutiny only the words of a specific structure, and focused on quantity degree features.

Therefore a need evolved, to measure for the Estonian language text-speech synthesis, the pauses and the boundary foot lengthenings, on the basis of a text read out from real speech, and to model their durations and locations in the speech flow.

2. Source material

Because we are concerned with text-to-speech synthesiser, the source material was a sample of texts read by announcers. Under condition that there is one-to-one conformity of text and speech, the symbol representation of prosody may be replaced by an acoustic representation, whereas it is possible to establish whether and for what extent the syntactic parsing of text is related to the prosodic parsing of speech.

Elected as the base material were:

- Passages of speech from the CD-version of a detective story (Stout 2003) read by an actor;
- Passages of speech and texts from the longer news from Estonian Radio, read by announcers;
- Passages of speech from the Estonian phonetic database BABEL (Eek, Meister 1998).

Altogether, 44 passages of speech were analysed (each 0.4-2 minutes long), in the presentation of 27 speakers (14 men and 13 women). All passages of speech were segmented into sounds and pauses.

3. Analysis of pauses and prepausal lengthenings

With a view to analysing the pauses and foot lengthenings, the durations of pauses derived from the speech wave were measured, and the foot lengthenings were calculated. For calculation of foot lengthenings, the durations of sounds comprising the foot were summed up, after which the actual duration was compared to the mean duration of the given foot structure in the speech of a concrete announcer. The first hypothesis was whether pauses and foot lengthenings could be classified (for instance, whether the pauses between phrases[2] differ significantly from the sentence end or paragraph end pauses). Presented in Table 1 are the mean durations of pauses as per announcers, the mean values for male and female announcers and the generalised mean. As is seen, the variance of even the mean values is very large. Curiously however, the generalised means of male and female pauses differ from one another as per durations within 10% only. The general mean visual observations suggest that in case of a text read out at normal speech rate the classification of speech pauses is fully possible. The statistical analysis of samples corroborates this surmise. The analysis in pairs of the logarithmic durations of pauses with the help of Student t-test reveals that the t-statistic values on significance level p=0,01 noticeably exceed the t-critical two-tail quantile (cf. Table 2). Hence it seems proved that the mean values of durations of pauses differ and the classification of pauses is fully possible, which fact could be applied in speech synthesis, for that matter.

[2] In this work, the phrase means the clause or element of enumeration, which has been determinated within the sentence by punctuation mark or conjunction.

Table 1. Durations of pauses and boundary lengthenings (ms) in speech

Dictors	*Phrase end pauses*	*Sentence end pauses*	*Paragraph end pauses*	*Phrase end lengthenings*	*Sentence end lenthenings*	*Paragraph end lengthenings*
Actor1 (m)	352	558	1025	200	220	315
Announcer1 (f)	303	828	902	124	112	117
Announcer2 (m)	286	769	1132	95	90	122
Speaker1 (f)		547		103	107	113
Speaker2 (m)	361	862		60	73	77
Speaker3 (f)	255	306		76	138	
Speaker4 (m)	145	478		78		89
Speaker5 (f)	275	879		109	100	88
Speaker6 (f)	470	1179		89	89	
Speaker7 (f)	117	559		85		64
Speaker8 (m)	416	829		73	118	75
Speaker9 (m)	260	683		94	91	
Speaker10 (m)	457	1210		93	73	
Speaker11 (f)	221	540		98	72	
Speaker12 (m)	144	667		114	95	77
Speaker13 (f)	148	429		122	75	
Speaker14 (m)	333	646		97	93	60
Speaker15 (m)	248	548		80		
Speaker16 (f)	379	621		123	116	75
Speaker17 (m)		700		79	101	
Speaker18 (m)	367	826		123	89	
Speaker19 (f)	342	945		74	104	
Speaker20 (f)	239	484		103	76	81
Speaker21 (f)	288	699		112	145	153
Speaker22 (m)	345	1023		101	85	
Speaker23 (f)	325	782		109	87	
Speaker24 (m)	398	721		122	90	120
Mean of female dictors	285	699	1132	97	103	111
Mean of female dictors	318	725	967	130	128	159
Generalised Mean	**302**	**715**	**1024**	**115**	**118**	**135**

When analysing, with the help of Student t-test the data of foot lengthenings (cf. Table 2) we had to stick to the zero-hypothesis: the foot lengthenings are from samples of the same mean value.

Taken under scrutiny next, was whether and to what extent the prosodic parsing of speech correlates with syntactic parsing where the latter is indicated by punctuation marks and conjunctions. As evidenced in Table 3, in speech the pause[3] is invariably at every paragraph end and sentence end. Two third of commas are connected with pauses. The least marked in speech are phrases starting with such co-ordinating conjunctions, which do not require the comma.

[3] We have treated as a prosodic pause, in this work the interruption of speech over 50 ms.

Table 2. Student t-test results for comparing of two-sample means (Ph-Se – between phrase and sentence, Ph-Pa – between phrase and paragraph, Se-Pa – between sentence and paragraph)

	Pauses			*Foot lengthenings*		
	Ph-Se	*Ph-Pa*	*Se-Pa*	*Ph-Se*	*Ph-Pa*	*Se-Pa*
T stat	16,06	20,06	8,00	0,61	0,71	0,39
T critical two-tail	2,59	2,64	2,71	2,60	2,73	2,72
P(T<=t)	<0,0001	<0,0001	<0,0001	0,54	0,48	0,70

From among the punctuation marks, it is the dash that the lengthening is clearly connected with. Apparently, it is the spell of the form of the mark – the long line - that makes reader drawl. Suggestive of the link between pauses and boundary lengthenings is the English term „prepausal lengthening“. The said term applies, on the basis of the Estonian language speech material, only in the extent of 60% (out of 601 pauses, the only 360 pauses displayed prepausal foot lengthening). According to perception tests carried out by Lehiste (Lehiste, Fox 1993) the Estonians expect a significantly lesser end lengthening on the last syllable of sentence that the English speakers do, as a matter of fact.

Table 3. Connection of pauses and foot lengthenings with the text parsing

	No of parsings in the text	*No of corresponding pauses in the speech*		*No of corresponding foot lengthenings in the speech*	
		Cnt	*%*	*cnt*	*%*
Paragrph end	21	21	100	15	71
Sentence end	185	185	100	124	67
Comma	179	120	67	94	53
Conjunction	85	42	49	46	55
Colon	11	10	91	6	57
Dash	15	13	87	14	93

4. Modelling of pauses and boundary lengthenings

Elucidation of mean durations of pauses and boundary lengthenings, in itself will not guarantee a natural rhythm of synthetic speech. On the basis of the analysis carried out in the previous section, in real speech the pauses and boundary lengthenings have a very large variance both in duration and also in their location in the speech flow. In order to preserve that variance in the synthetic speech, to some extent at least, the temporal structure of the speech must be modelled, according to context.

4.1. Modelling of durations

For modelling the pauses and boundary lengthenings, a vector of argument features (explanatory variables) consisting of 18 features was generated, basing on the text. Lumped under the vector were those features or factors that were likely to impact on duration of pause or foot lengthening.

The features give an indication of the structure of the text (e.g. sentence or phrase end, phrase length), prepausal or lengthened foot (e.g. foot quantity, foot length, length of last syllable of foot) and temporal structure of the speech (distance from the previous pause, distance from the intonation phrase). The response was the logarithmic duration, which normalises the distribution of durations. The general parameters of the final model have been presented in Table 4. Quite clearly the model is significant and it described a large part of logarithmic duration variance (R-square=0,5564). Significant features impacting on pause duration are those related to punctuation marks and conjunction. A significant feature too is the distance from the previous pause and whether the foot preceding the pause has been lengthened. Construction of prediction model to pauses is seemingly successful, however modelling of foot lengthenings in the speech flopped, because the links with features were weak and the model described a small part only (R-square=0,1102) of variance of duration of lengthenings.

Table 4. Summary of fit and the analysis of variance for the regression model of durations

<table>
<tr><td colspan="6"><i>Summary of Fit</i></td></tr>
<tr><td colspan="3">Mean of Response -0,86673</td><td colspan="3">R-Square 0,5564</td></tr>
<tr><td colspan="6">Analysis of Variance</td></tr>
<tr><td>Source</td><td>DF</td><td>Sum of Squares</td><td>Mean Square</td><td>F Stat</td><td>Pr &gt; F</td></tr>
<tr><td>Model
Error
C Total</td><td>6
560
566</td><td>272,2
216,9
489,1</td><td>45,36
0,39</td><td>117,07</td><td>&lt;.0001</td></tr>
</table>

4.2. Modelling of situation of pauses in speech

When modelling the situation of pauses, we used the selection of argument features similar to that used in modelling the durations. Besides that we supposed that situation of the pauses may be in correlation with proper names and foreign words. We presumed that in front of the more complicated words in the speech flow, there could be a pause, and that after proper names we would tend to make interruptions in the speech (à la '*My name is Bond, James Bond*'). Anticipatively, that hypothesis failed to find proof on the basis of the given material. Response of the model was the value of probability that a pause would be made after a certain word.

$$P(PAUS)= -4.91 + 2{,}30 * FRKOM + 1{,}46 * FRSID + 3{,}80 * FRKLMK + 0{,}07 * KAUGLA + 0{,}21 * TAKTVXLDE + 0{,}20 * TAKPIKHX + 2{,}19 * PIKENDUS$$

Figure 1. Model equation (P(PAUS) – probability value for a pause in the speech flow, FRKOM – phrase end (comma), FRSID – phrase end (conjunction), FRKLMK – phrase end (colon or dash), KAUGLA – distance from beginning of sentence in foots, TAKTVXLDE – quantity degree of the last foot, TAKPIKHX – length of last foot in sounds, PIKENDUS – feature of the lengthening of last foot)

The form of this binominal model has been presented in Figure 1. Here, too the situation of pause is in strong correlation with punctuation marks and co-ordinating conjunction. The probability of interruption of the speech flow is also heightened by distance from the beginning of sentence and whether the last foot was lengthened, as well as the last foot length and the quantity degree of the foot. The duration of the pause will be calculated with the duration model found in p. 4.1. Admittedly prediction of lengthenings flopped, again. The model turned out inadequate.

5. Summary

Analysed in this work was the comportment of pauses and boundary lengthenings in the read out speech. In regression analysis, simple models for prediction of duration of pauses and their location in synthetic speech were found. For prediction of boundary lengthenings, no reliable prediction model could be derived. To all appearances, the lengthenings should not be treated in an isolated manner; rather they should be viewed as part of prediction model of sounds (Mihkla, Pajupuu, Kerge, Kuusik 2004). In further work, when composing the prediction models of pauses, different statistical methods too, should be used (e.g. neuron nets).

References

Eek, Arvo; Meister, Einar 2003. Foneetilisi katseid ja arutlusi kvantiteedi alalt (I): Häälikukestusi muutvad kontekstid ja välde. In: *Keel ja Kirjandu,* 11, 815–837.

Eek, Arvo; Meister, Einar 1998. Estonian Speech in the Babel Multilanguage Database: Phonetic-Phonological Problems Revealed in the Text Corpus. In: *Proceeedings of the Workshop on Speech Development for Central and Eastern European Languages.* The First International Conference on Language Resources and Evaluation, Granada.

Krull, Diana 1997. Prepausal lengthening in Estonian: Evidence from Conversational speech. In: Lehiste, I.; Ross, J. (eds.). *Estonian Prosody: Papers from a Symposium, Proceedings of the International Symposium on Estonian Prosody, Tallinn, Estonia, October 29-30, 1996*. Institute of the Estonian Language and Authors, Tallinn. 136–148.

Lehiste, Ilse 1981. Sentence and paragraph boundaries in Estonian. In: *Congressus Quintus Internationalis Fenno-Ugristarum, Turku, 20.-27. 1980, Pars VI*. 164-169.

Lehiste, Ilse; Fox, Robert 1993. Influence of duration and amplitude on the perception of prominence by Swedish listeners. In: *Speech Communication* 13, 149–154.

Mihkla, Meelis; Pajupuu, Hille; Kerge, Krista; Kuusik, Jüri 2004. Prosody modelling for Estonian text-to-speech synthesis. In: *The first baltic conference on human language technologies: the baltic perspective. Riga, April 21–22 2004,* Riga. 127–131.

Stout, Rex 2003. *Deemoni surm.* CD-versioon (loeb Andres Ots). Tallinn: Elmatar.

Zvonik, Elena; Cummins, Fred 2002. Pause duration and variability in read texts. In: *Proceedings of the 7th International Conference on Spoken Language Processing,* Denver, ICSLP-2002. 1109–1112.

MEELIS MIHKLA is assistant director, Institute of the Estonian Language, Tallinn. He received his M.A. (Estonian language) at University of Tartu, dealing with Estonian text-to-speech synthesis. His research interests concern prosody modelling and speech units databases. His doctoral study focuses on statistical modelling of temporal structure of speech.

ESTSUM – ESTONIAN NEWSPAPER TEXTS SUMMARIZER

Kaili Müürisep, Pilleriin Mutso
University of Tartu, Estonia

Abstract

This article describes an experimental software system for automatic summary generation of Estonian newspaper texts called EstSum. EstSum constructs short summaries of text by selecting the key sentences that characterize the document. Sentences are ranked for potential inclusion in the summary using a weighted combination of statistical, linguistic and typographic features like the position, format and type of sentence, and the word frequency. During the testing, a corpus of 10 hand-created summaries of neswpaper articles was used. The summarizer's output was compared to the handmade summaries and the percentage of overlapping sentences was 60% in average.

Keywords: summarization, Estonian language

1. Introduction

As the amount of on-line information increases, more and more effort is dedicated to creating automatic summarization systems. Since the automatic text summarization is largely a language-specific task, suitable algorithms must be found for each natural language. This paper describes our summarization work for Estonian.

According to Radev et al, a summary can be loosely defined as a text that is produced from one or more texts, that conveys important information in the original text (s), and that is no longer than half of the original text(s) and usually sicnificantly less than that. In other words, the main goal of a summary is to present the main ideas in a document in less space (Radev et al. 2002).

Most research on summary generation techniques still relies on extraction of important sentences from the original document to form a summary. There are several methods for measuring the importance of a sentemce. Some algorithms calculate a weight for each sentence, taking into account the position of the sentence and word frequencies (Dalianis et al. 2003), while other algorithms use semantic information (e.g. WordNet), in order to find the hierarchy of concepts.

There are also different methods for summary generation from a single document and from multiple documents.

Summarization tool for Estonian (EstSum) focuses on extraction methods from a single document. Also the area of texts is limited: EstSum considers that the input text is formed as news text.

EstSum can be considered as the first tool for automatic summarization adapted for Estonian. It should be noted that there exist some summarization systems that are language-independent, with MS Office AutoSummarizer probably being the best known example (the algorithm behind it has not been publicly released). SweSum is another well-known language-independent summarizer which is also publicly available on the Internet (Dalianis 2000).

2. Overview of EstSum

EstSum has been written in Perl language, and it consists of three modules: HTML converter, sentence splitter and extractor.

HTML converter removes unimportant tags, normalizes the crossing labels and converts input to SGML format. It marks the headers and subheaders using font information, gives special labels to captions of photos and removes tables. It also preserves the important information about font, distinguishing between bold, italic and default font.

Sentence splitter is uses the rule-based approach for processing its input, employing 30 rules that consider the different cases of sentence beginnings and endings.

EstSum has two options for calculating text compression rates. With the first option, EstSum considers sentences as units, and when the text of 100 sentences is compacted by 30%, the generated summary has 30 sentences. With the second option, words are considered as units that sometimes helps to exclude long sentences from the summary.

EstSum extracts salient sentences from the text using location, format and keyword based information about sentence. The overall method of scoring sentences for extraction is based on a linear function of the weights of each of the three features, similar to Edmundson's style formula (Edmundson 1969; Mani 2001):

(1) $W(s)=\alpha P(s)+\beta F(s)+\gamma K(s)$

Here W(s) is the weight function of sentence s, P(s) is the position-based score function, F(s) the format-based score function and K(s) the keyword-based score function; α, β and γ are constants.

The feature weights and tuning parameters α, β and γ have been adjusted by hand using a manually created training corpus of extracts and the knowledge of authors.

The coprus of extracts is relatively small (only 20 texts) for drawing any final statistical conclusions. Despite this the main tendencies for selecting salient sentences are detectable. The length of extracts is 30% of length of the original texts. The smallest original text contained 4 sentences and the largest one 41 sentences, with the average text length being 18 sentences. The texts belonged to various genres - short news, columns, feature stories and one interview.

2.1. Position-based scoring

Position-based scoring considers the sentence location. In order to find appropriate weights for position-based scoring, we investigated how the summaries in the training corpus reflect the first 3 sentences of the original text, the first sentence after each subtitle and the first 2 sentences of each paragraph.

We established that the most influential sentences are the sentences following the title – the first sentence of the text was included in the summary in 100% of the cases, the second and the third sentence in 65% of the cases. The sentences immediately following the subtitles were included in the 60% of the cases.

We also found that the first sentence of the paragraph was included in the summary in 40% of the cases, and the second and the third in 20% of the cases. In addition, 20% of the summaries contained the last sentence of the text. The position-based scores are given in Table 1.

Table 1. Position based scores

Feature	Percentage in extracts	Given score
1st sentence in article	100	10
2nd sentence in article	65	7
3rd sentence in article	65	7
1st sentence after subheader	60	6
1st sentence in paragraph	40	4
2nd sentence in paragraph	20	2
3rd sentence n paragraph	20	2
Other	6	0

The scores are normalized using formula (2).

$$(2) \quad n = \frac{p \cdot 100}{t}$$

Here *n* is normalized score, *p* is assigned score of the sentence and *t* is total of all position scores in the article.

2.2. Format-based scoring

Format-based scoring considers the sentence font (default, bold or italic) and punctuation marks (exclamation and question marks, double quotes). Figure captions and the text author are also detected and given minimum scores. Table 2 depicts the features and scores.

Table 2. Format based scores

Feature	Percentage	Given score
Default font	32	3
Bold or italic	70	10
Question or exclamation mark in sentence	10	0
Quotation marks in sentence	18	2
Captions, authors, subheaders	0	0

The scores are normalized using same algorithm as formula (2).

2.3. Keyword-based scoring

The first version of EstSum did not use any linguistic modules, so it was possible to use only word forms instead of roots.

Keyword-based scoring uses two techniques for detecting keywords: finding words that are relatively frequent in this article and not very frequent in general word frequency table; extracting words from the text title and all subtitles.

However, when inspecting the training corpus, we discovered that only 48% of the sentences containing words from the titles were included in summaries. Also, if extra score is assigned to sentences containing most frequent words, then only 25% of the sentences with highest scores are actually present in summaries. Therefore, when discovering frequent word forms, the summarizer must also employ a general word frequency table for a given language, in order to estimate whether the word form appears more frequently than it normally does in texts written in that language.

Our keyword-based scoring algorithm also uses a general word frequency table that is generated from the newspaper texts of 400,000 words. The table lists word frequencies per 10,000 words and contains 1100 words that occur at least once in texts of 10,000 words.

The words belonging to the title (article headline) and subtitles are given extra scores. (5 and 2 points respectively). All the other words are put into the local frequency table with a weight 1.

2.4. Tuning general parameters

In order to tune general parameters, we measured how many of the sentences in summaries are found by applying each weight function separately. The position-based weight function assigned high scores to the first 3 sentences of the text, and the format-based function assigned high scores to the first 1-2 sentences of the text, while the rest of the sentences received relatively similar scores from all methods. Since the position- and format-based function yielded better results, we decided to use 0.4 as coefficient for them in the formula (1), while the coefficient for the keyword-based function was set to 0.2.

With these settings, 51% of the sentences present in the training corpus are also chosen by the EstSum summarizer for inclusion in the summaries (the figure of 51% does not reflect the text titles which belong to summaries by default).

3. Evaluation

Evaluating automatically generated summaries is not a straightforward process. The evaluation is usually made by comparing automatically generated summaries to summaries compiled by humans. Such evaluation gives good results in other domains of language technology like tagging and parsing, but sentence selection for summary is not so well defined task, and the summaries may be subjective depending on the author's interests and the mood of the moment. Hassel (2003) has found that at best there was a 70% agreement between summaries generated by two individuals, Radev and others have reported a figure of 60% (Radev et al. 2002).

3.1. Corpus for evaluation

The small corpus for evaluation consists of 11 texts with the average length of 321 words and 23 sentences. These texts are more uniform by their genre (front page stories, domestic news, business news and sports news from one newspaper).

3.2. Results

EstSum was able to choose 60% of sentences from the evaluation corpus as an average. In the best case the figure was 85.7% and in the worst case it was 0%. In the latter case the text was a very short newspaper article and EstSum chose the article title for the summary, while the manually compiled summary was longer than 30% of the words.

3.3. Comparison with other tools

SweSum is a freely available[1] summarizer that has some common features with EstSum. SweSum has been desinged for the processing newspaper text, and thus it uses so called position score: the sentences in the beginning of the text are given higher scores than the ones in the end. HTML tags which indicate sentences with bold text are given a higher score than the ones without bold text tagging, dito title tagging. Sentences containing numerical data are given a higher score than the ones without numerical values. Sentences which contain keywords are scored high. SweSum has a linguistic module for Swedish texts that finds the stem of each word. For Estonian, we used SweSum with a generic language option. All the above parameters were normalized and put in a naïve

[1] http://swesum.nada.kth.se/

combination function with no special weighting to obtain the total score of each sentence. (Dalianis 2000). SweSum without the linguistic module selected 41% of the sentences from the manually created test corpus.

Since 1997, the Microsoft Word editor has also a summarizer for documents. Unfortunately, it performs rather poorly on Estonian texts. For example, in a number of cases the summarizer is unable to detect sentence boundaries. During our experiments we found that approximately 25% of the extracted sentences were same as in the benchmark corpus.

4. Conclusions and future extensions

The automatic text summarizer tool EstSum presented in this paper can still be regarded as a prototype and is thus rather actively developed. Although the preliminary results by EstSum are excellent when compared to other two systems described in this paper, the evaluation was carried out on relatively small data sets, and therefore EstSum needs a considerable amount of further development and testing. Our current plans include the addition of a linguistic module to the EstSum framework for morphological analysis and morphosyntactic disambiguation. The employment of the linguistic module would make the keyword detection more efficient. We also plan to work on pronoun resolution, in order to make the summarized text more coherent.

Apart from developing EstSum, another important task is the creation of a larger training and test corpus that could be used for advanced statistical analysis and machine learning. The methods of measuring the summarizer performance are also a subject of further research.

References

Dalianis H. 2000. SweSum – A text summarizer for Swedish. *Technical report TRITA-NAP0015, IPLab-174*, NADA, KTH. October 2000. Retrieved February 27, 2004, from http://www.nada.kth.se/~hercules/Textsumsummary.html

Dalianis, H., M. Hassel, J. Wedekind, D. Haltrup, K. de Smedt and T.C. Lech. 2003. Automatic text summarization for the Scandinavian languages. In Holmboe, H. (ed.) *Nordisk Sprogteknologi 2002: Årbog for Nordisk Språkteknologisk Forskningsprogram 2000-2004.* Museum Tusculanums Forlag. 153-163.

Edmundson, H.P. 1969. New methods in automatic abstracting. In: *Journal of the Association for Computing Machinery 16 (2).* 264-285. Reprinted in: Mani, I.; Maybury, M.T. (eds.) *Advances in Automatic Text Summarization.* Cambridge, Massachusetts: MIT Press. 21-42.

Hassel, M. 2003. Exploitation of Named Entities in Automatic Text Summarization for Swedish. In the *Proceedings of NODALIDA '03* - 14th Nordic Conference on Computational Linguistics, May 30-31 2003, Reykjavik, Iceland.

Mani Inderjeet 2001 Automatic summarization. Amsterdam: John Benjamins Publishing Co.

Radev D. R., E. Hovy, K. McKeown. 2002. Introduction to the special issue on summarization. *Computational Linguistics.* Vol 28(4). 399-408.

KAILI MÜÜRISEP is a researcher at the Institute of Computer Science, University of Tartu. She received her Ph. D. (computer science) at the University of Tartu, dealing with automatic syntactic analysis of Estonian. Her research interests concern automatic syntactic analysis of written and spoken language, treebanks and automatic summary generation. E-mail: kaili.muurisep@ut.ee.

PILLERIIN MUTSO is a master student at the Institute of Computer Science, University of Tartu. Her research interests concern automatic summary generation and the evaluation of generated summaries. E-mail: pmutso@ut.ee.

APPLYING MFCC-BASED AUTOMATIC SPEAKER RECOGNITION TO GSM AND FORENSIC DATA

Tuija Niemi-Laitinen*, Juhani Saastamoinen, Tomi Kinnunen**, Pasi Fränti****
*Crime Laboratory, NBI, Finland
**Dept. of Computer Science, Univ. of Joensuu, Finland

Abstract

Speaker Profiler computer program for automatic speaker recognition has been developed in a research project funded by the Finnish Technology Agency. A vector quantization (VQ) matching approach is used, where dissimilarity of an unknown speech sample is computed for codebooks created using the K-means algorithm. This study tests the recognition reliability with two databases constructed from Finnish band-limited GSM speech and authentic crime case speech.

Material for the first test is recorded with a GSM phone and a laptop computer. Spontaneous speech vs. reading was tested. The program should pick the right person from the database based on independent non-verbatim speech samples. There were 47.5 % out of 107 samples ranked first correctly. Some very poor quality speech files were used in training and the mother tongue for some speakers was not Finnish. If these samples were not considered, the result was better.

The second part of this study consists of real crime investigation cases. The speaker database was constructed from known speech samples (suspect). Unknown sample(s) recorded at the crime scene were matched against the database. From the matched 61 samples, 68.9 % were ranked first correctly. Accuracy is sufficient for creating shortlists in forensics.

Keywords: speaker recognition, forensics, automatic recognition, real-time recognition, MFCC

1. The aim of the study

The aim of the study is to test how reliably an automatic speaker recognition program performs when the input speech is band-limited (GSM speech) and authentic (real crime case material). Test situation where speech files have been recorded in good laboratory conditions with high quality microphones is far from the reality in forensics. This is why GSM phone recordings were used for this study. Annually most speaker identification cases at the Crime Laboratory consist of GSM speech material only. The tests simulate a typical speaker recognition case. Usually the speech of a criminal (the unknown sample) is spontaneous. The speech of the suspect, on the other hand, usually consists of both reading and semi-spontaneous speech.

2. Speech material and recordings

Table 1 lists some statistics describing the overall structure of the data sets used in this study. These figures are explained in detail in sections 2.1 and 2.2.

Table 1. The number of samples and their duration in GSM and Forensic data

Data		Number of speech samples			Sample duration (sec.)		
Test set	*Subset*	Male	Female	Total	Min.	Max.	Avg.
GSM	Read	47	60	107	140	252	183
GSM	Spont.	47	60	107	48	300	153
GSM	Both together	94	120	214	48	300	168
Forensic	Suspect	27	1	28	6	189	73
Forensic	Crime scene	59	2	61	2.4	379	50
Forensic	Both together	86	3	89	2.4	379	57

2.1. GSM data

Recordings for the first part of the tests were collected during 2001 under the project "The Joint Project Finnish Speech Technology" supported by the National Technology Agency (TEKES agreements 40285/00, 40406/01, 40238/02) and titled "Speaker Recognition" (University of Helsinki Project No. 460325).

For the present study, 107 speakers (60 female and 47 male) were recorded using a GSM phone and portable computer were selected. The samples included 16 noisy recordings as well as 12 samples spoken by persons whose mother tongue was Estonian, Russian or Swedish. The duration of the samples in spontaneous speech varied from 48 to 300 seconds, and in text reading task from 140 to 252 seconds.

2.2. Authentic crime case data

Real crime investigation speech data is used in the second part of the study. The known and unknown speech samples in several crime investigation cases were recorded during the years 2000-2004 either via GSM or land-line phones. The phone type could not be limited to one, because the criminals use the phones they have. One speaker (3 samples) is a female, the others are male. The female has very low-pitched voice, and she was included for testing reasons. Only speech files not containing extra background noise, were considered in the test. Some of the files did still contain some noise, e.g. computer hum and traffic sounds. Total of 61 unknown and 28 known samples were used. The durations of the samples varied from few seconds to 379 seconds.

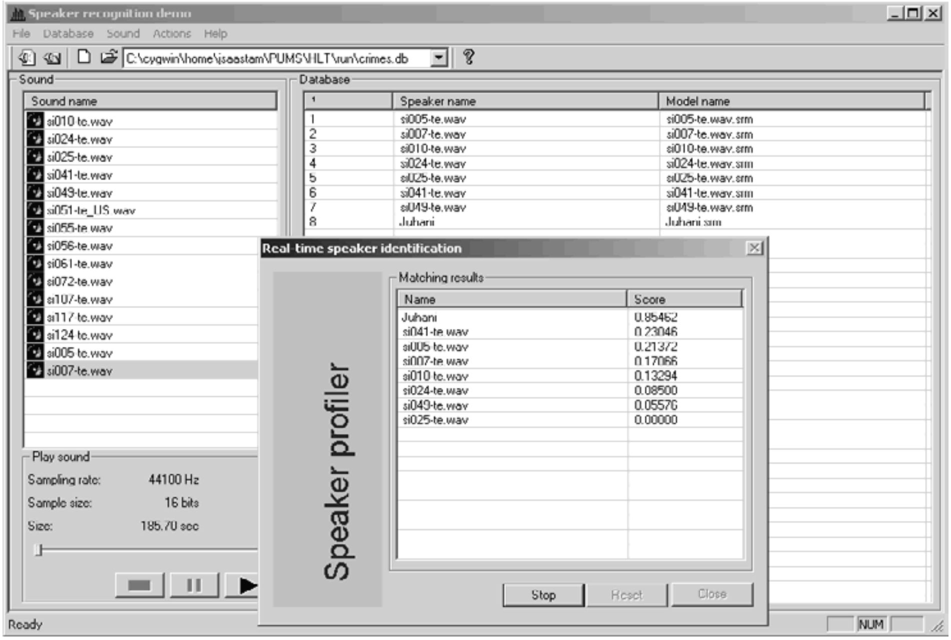
3. Winsprofiler speaker recognition software

The *Speaker Profiler* (*sprofiler*) is a portable software engine for creating, managing, and recognising voice profiles based on speech samples. The *Winsprofiler* program used in the tests of this study, is a realization of sprofiler that is augmented by a Windows graphical user interface. It supports both training and recognition from sound files or directly from PC-microphone, as well as drag-and-drop input of files. Recognition is fully automated, and the software supports both offline matching from previously recorded samples, as well as real-time matching with speech input stream from microphone of PC-soundcard. The software was developed in a research project funded by the Finnish Technology Agency (TEKES agreements 40437/03 and 40398/04).

Speaker Profiler uses mel-frequency cepstral coefficients (MFCC) as the acoustic features. We have compared different spectral features for automatic speaker identification on several databases, including subband analysis, mel- and Bark-warped cepstra, LPC-based features and formant frequencies, along with their delta features (Kinnunen *& al.* 2004a, 2004b). MFCC's have shown high accuracy in our experiments for both laboratory- and telephone-quality speech.

Speaker profiler uses 12 lowest MFCC coefficients, excluding the 0th coefficient, which is not a robust parameter as it depends on the intensity. The filterbank employed in the MFCC computation consists of 27 triangular filters equispaced on the mel frequency scale. A frame rate of 100 frames per second is used, with a 20 ms overlap between the adjacent frames (30 ms frame length, 10 ms frame shift).

Speaker matching is based on a vector quantization (VQ) approach (Kinnunen *& al.* 2004c). For each enrolled speaker, a codebook of size 64 is created using the K-means algorithm. During the matching, an unknown sample is scored against the stored codebooks, giving a dissimilarity value for each speaker. For easy interpretation, the scores are normalized into the interval [0,1] so that a larger score means better match. The program displays a ranked list of similarity values. Figure 1 shows a Winsprofiler screenshot. *Juhani* is speaking to a microphone (connected to PC-soundcard). Feature vectors computed from the speech are scored on-line against the database of 8 models, 7 were trained using sound files. Juhani was trained using the PC-soundcard.



Sound name
si010 te.wav
si024-te.wav
si025-te.wav
si041-te.wav
si049-te.wav
si051-te_US.wav
si055-te.wav
si056-te.wav
si061-te.wav
si072-te.wav
si107-te.wav
si117 te.wav
si124 te.wav
si005 te.wav
si007-te.wav

Sampling rate: 44100 Hz
Sample size: 16 bits
Size: 185.70 sec

1	Speaker name	Model name
1	si005-te.wav	si005-te.wav.srm
2	si007-te.wav	si007-te.wav.srm
3	si010-te.wav	si010-te.wav.srm
4	si024-te.wav	si024-te.wav.srm
5	si025-te.wav	si025-te.wav.srm
6	si041-te.wav	si041-te.wav.srm
7	si049-te.wav	si049-te.wav.srm
8	Juhani	Juhani.srm

Name	Score
Juhani	0.85462
si041-te.wav	0.23046
si005 te.wav	0.21372
si007-te.wav	0.17066
si010 te.wav	0.13294
si024-te.wav	0.08500
si049-te.wav	0.05576
si025-te.wav	0.00000

Figure 1. *Winsprofiler* real-time matching, *Juhani*'s speech is recorded and matched on-line against 8 database entries, including *Juhani*'s voice model

4. Test results with GSM data

In the first test situation spontaneous speech vs. reading was tested. Here the question was, does the program find the right person from the database when his/her two different, non-verbatim, speech samples are compared. First the database was created from the spontaneous speech samples and then, for reliability reasons, the database was also created from the text reading samples.

When the speaker database was constructed from the spontaneous speech samples, and the text reading samples were matched, there were 51 out of 107 tested samples ranked first (47.7 %), 68.2 % ranked among the first 3 samples, and 76.6 %

among the first 5 samples. The database contained 16 samples that were somewhat noisy or the voice of the speaker was either creaky or had very low pitch. After removing these 16 samples from the database, total of 91 samples formed a new database. The upper part of Table 2 shows the results of this test. When both noisy and nonnative speakers samples were removed, 53.2 % were ranked first, 73.4 % were ranked among the first 3 samples, and 81 % among the first 5 samples.

The speaker database was also formed from the text reading samples to check the reliability of the results. Total of 107 samples formed the database. When spontaneous speech samples of the same speakers were used as the reference, there were 25 samples out of 107 tested ranked first (23.4 %), 39.3 % was ranked among the first 3 samples, and 51.5 % among the first 5 samples, see Table 2 (lower part). When noisy and nonnative speakers' samples were removed, 62 % were ranked first. 76 % were ranked among the first 3 samples, and 87.3 % among the first 5 samples (see Table 2).

Table 2. Rankings of the correct speaker. The database is constructed from spontaneous speech samples and the testing material from text reading of the same speakers (upper part), and vice versa (lower part)

Read speech		Rank of the correct speaker		
	# Samples	1	1-3	1-5
All samples	107	47.7 %	68.2 %	76.6 %
Noisy samples removed	91	49.5 %	70.3 %	80.2 %
Noisy + non-native removed	79	53.2 %	73.4 %	81.0 %
Spontaneous		Rank of the correct speaker		
	# Samples	1	1-3	1-5
All samples	107	23.4 %	39.3 %	51.5 %
Noisy samples removed	91	57.1 %	73.6 %	83.5 %
Noisy + non-native removed	79	62.0 %	76.0 %	87.3 %

Poor quality of some of the recordings lowered the recognition score and the ratings. Noisy samples could affect many matching situations, where the recognition is not clear. Some tested speech samples were uttered in Finnish, but the speaker's mother tongue is Russian, Estonian, or Swedish (many Finns have Swedish as their mother tongue). This had some effect on the results. The reading and spontaneous speech samples of these speakers did not match as often as the others'.

5. Test results with real crime data

The second part of this study consists of real crime investigation cases. The known speech samples (suspect) form a speaker database. The unknown speech sample(s) recorded at the crime scene, were used as testing material. The test was done also vice versa: the unknown crime scene speech samples formed the speaker database and the known speech sample(s) from the suspects were used as testing material.

In the first case, 28 known speech samples from suspects formed the database, and 61 different phone calls from 13 different criminal cases were matched against the database. All the speech material in the criminal cases was recorded via GSM or land-line phones. From the matched 61 samples, 68.9 % were ranked first, 82 % among the first three, and 85.2 % among the first five. Three cases had either very short samples or the samples were noisy. These results are shown in Table 3 separately.

The testing with the forensic data was done twice. When the database consisted of 68 unknown criminal speech samples from 13 different crime investigation cases, and the test material consisted of 25 known speech samples from suspects, the result was much worse. Only few samples of suspects and criminals matched. This situation is not typical. Databases consisting of unknown speakers are not used in forensics.

Table 3. Rankings of the correct speakers and the corresponding correct identification rates. The database is constructed from the 28 known samples (upper part) and from the 68 unknown samples (lower part)

Known samples		Rank of the correct speaker		
	# Samples	1	1-3	1-5
All samples	61	68.9 %	82.0 %	85.2 %
Noisy and short samples removed	58	72.4 %	86.2 %	89.7 %
Unknown samples		Rank of the correct speaker		
	# Samples	1	1-3	1-5
All samples	25	40.0 %	52.0 %	68.0 %
Noisy and short samples removed	22	45.5 %	59.0 %	72.7 %

6. Conclusion

There were some interesting findings in the spontaneous and text reading tests. The creaky voiced female speakers were often recognized as someone else. Likewise, the males with very low-pitched and/or creaky voice succeeded similarly. Some speakers in the database were ranked first many times when the samples from another speakers were matched against the database. The reason for this should be studied.

All the speech material in the criminal cases (test-2) was recorded via GSM or land-line phones and no extra background noise was found. This could be an error source for the study. Still the question arises, does the matching use more information from the background than from the speech signal itself.

Different results arise with the two opposite forensic tests, though the same samples are used in both. When the database is formed from known, usually long speech samples, the result is better than with the database consisting of unknown speech samples that are usually quite short. Is the sample duration the reason for this problem? The database sizes were different also, the recognition result was worse for the larger.

Currently the *Speaker Profiler* implements a baseline MFCC-VQ recognition. According to our experience, is quite accurate for matched conditions. However, more accuracy is desired in acoustically mismatched conditions, such as technical mismatches in recording. Further studies are therefore needed to detect the criticial conditions, and to employ methods for noise-, channel-, or score normalization.

The recognition performance of the Speaker Profiler is nowhere near perfect. However, it can be already as such used to create *shortlists,* i.e. pick substantially smaller sets of best ranked speaker candidates. The correct speaker is consistently found in the shortlist. This property already makes it a valuable tool for a crime investigator.

Acknowledgements

The Joint Project of Finnish Speech Technology was supported by TEKES, Civil Aviation Administration in Finland, The Finnish Defence Forces, National Bureau of Investigation in Finland, Accident Investigation Board Finland and Scando Oy.

References

Alexander, A., Botti, F., Dessimoz, D. And Drygajlo, A. 2004. The effect of mismatched recordings on human and automatic speaker recognition in forensic applications. *Forensic Science International* 146S (2004) S95-S99.

Botti, F., Alexander, A. And Drygajlo, A. 2004. On compensation of mismatched recording conditions in the Bayesian approach for forensic automatic speaker recognition. *Forensic Science International* 146S (2004) S101-S106.

Broeders, A.P.A. 1995. The role of automatic speaker recognition techniques in forensic investigation. *Proceedings of the XIIIth International Conference of Phonetic Sciences,* vol. 3, 154–161. Stockholm: KTH & Stockholm University.

Iivonen, A., Harinen, K., Keinänen, L., Kirjavainen, J., Meister, E. & Tuuri, L. 2003. Development of a multiparametric speaker profile for speaker recognition. *15th Int. Congress of Phonetic Sciences, Barcelona* 3-9 Aug, 2003, 695–698.

Kinnunen, T. (2004a). *Spectral features for automatic text-independent speaker recognition.* Licentiate's Thesis, Dept. of Computer Science, Univ. of Joensuu.

Kinnunen, T., Hautamäki, V., and Fränti, P. (2004b). "Fusion of spectral feature sets for accurate speaker identification", *Proc. 9th Int. Conference Speech and Computer* (*SPECOM'2004*), pp. 361-365, St. Petersburg, Russia, September 20-22, 2004.

Kinnunen, T., Karpov, E., and Fränti, P. (2004c). Real-time speaker identification and verification. Accepted for publication in *IEEE Transactions on Speech and Audio Processing*.

Künzel, H., Gonzáles-Rodríguez, J. 2003. Combining automatic and phonetic-acoustic speaker recognition techniques for forensic applications. *15th International Congress of Phonetic Sciences, Barcelona* 3-9 Aug, 2003, 1619–1622.

Niemi-Laitinen, T. 2001. Automatic speaker recognition – is it possible or not? In S. Ojala & J. Tuomainen (eds) *Papers from the 21st Meeting of Finnish Phoneticians* – Turku 4-5.1.2001. Publications of the Department of Finnish Language and General Linguistics, University of Turku 67: 71-80.

TUIJA NIEMI-LAITINEN is a researcher at the Crime Laboratory at the National Bureau of Investigation, Finland. Her research interests concern speech prosody and speaker recognition. E-mail: tuija.niemi@helsinki.fi or tuija.niemi-laitinen@krp.poliisi.fi

JUHANI SAASTAMOINEN is a project manager in the Department of Computer Science in the University of Joensuu, Finland. His current research interest is numerical methods in speech analysis. E-mail: juhani@cs.joensuu.fi

TOMI KINNUNEN is a doctoral student in the department of Computer Science in the University of Joensuu. His research topic is automatic speaker recognition. E-mail: tkinnu@cs.joensuu.fi

PASI FRÄNTI is a professor of Computer Science in the University of Joensuu, Finland. His primary research interests are in image compression, pattern recognition and data mining. E-mail: franti@cs.joensuu.fi

ANALYSIS OF LITHUANIAN SPEECH SOUND LENGTHS AND PITCH FOR PROSODY GENERATION

Ingrida Radziukynienė, Aušra Šurkutė, Asta Kazlauskienė, Minija Tamošiūnaitė
Vytautas Magnus University, Kaunas, Lithuania

Abstract

Sound duration and pitch are analyzed, based on three speakers continuous speech corpus. Varying speech rate from slow to medium (normal), and from normal to fast, length of both vowels and consonants varied significantly, but sounds in root and prefix varied less than sounds of the suffix and the ending. Pitch was investigated at the level of syllables. Rising pitch was prevailing in the stressed syllables, while falling was prevalent in the syllable following the stressed one. More complex models of pitch variation inside a single syllable, containing two or three linear parts, were analyzed as well, showing consistent patterns for stressed and post-stress syllables. Preliminary model for Lithuanian pitch generation at the syllable level was proposed.

Keywords: sound duration, duration variability, pitch model, speech rate, speech synthesis.

1. Introduction

Pitch and duration are widely acknowledged to be the most important prosodic characteristics of a language, and are required for qualitative speech synthesis (Keller 2002). Numerous phonetic studies of Lithuanian speech have revealed large amount of patterns, both for duration and pitch, at the level of a sound and a syllable, yet the focus of these investigations has usually been specialized, and data of the extent needed for prosody generation in speech synthesis applications does not exist.

Duration of Lithuanian vowels has been relatively widely investigated. Investigations consider topics like variability of vowel duration depending on stress or phrase accent, and context of the vowel in a word or phrase (Pakerys 1982, Kazlauskienė 2000). Less is known about diphthong duration, and consonant duration is investigated the least. Yet duration variability among individual speakers (and dialects) is so substantial, that for synthesis purposes using statistics of a corpus of a single speaker turns up to be a more reliable solution, than using averages from linguistic knowledge. However, the complication of tempo variability persists even for a single speaker. In other language analysis has been observed that various parts of the word do not vary proportionally with speech rate. In (Jance 2003) is found that consonant duration varies less than vowel duration, and in (Chung 1999) is found that root duration behaves in proportion with word length, while the suffix shortens more than the prefix in English words. Yet these types of regularities are expected to differ in different languages due to different functionalities of affixes. In our duration research we will address the question, to what degree proportionality of vowels and consonants positioned in various parts of the word are preserved in Lithuanian with varying speech rate. Those results are expected to contribute to normalization of duration information from an extensive single speaker corpus.

Characteristics of Lithuanian pitch have mainly be investigated in relation to the intonation of a sentence, or at the syllable level with the focus of accentuation issues (Pakerys 1982). Lithuanian stressed syllables may be accentuated in three different ways: grave, acute, and circumflex. Yet three classes of accent could not be reliably attributed to distinctive pitch patterns (Vaitkevičiūtė 1995: 89-91). This study investigates pitch properties only at the syllable level. Following (Pakerys 1982, Botinis 2001) we hypothesize that pitch is rising in the stressed syllable, and falls off later, in the same or the next syllable, possibly in relation with accentuation. Models with division of a syllable pitch into several linear parts are considered.

2. Data

Two texts of approx. 100 words each, red at three different rates by three speakers (two women and a man) were analyzed. Rate normalization was performed as follows: first the speaker was asked to read the text at his or her normal rate, then was asked to read the text at the quicker than the normal, and at the slower than the normal rates. Obtained speech epochs were manually segmented into sounds (for duration analysis) and syllables (for pitch analysis). Pitch was analyzed for two speakers normal rate.

3. Duration analysis

Sound lengths variations were analyzed on the transitions from quick to normal, and from normal to slow rate. Two additional factors were considered: sound position in the word (root, prefix, suffix, or the ending), and type of the sound (vowel, diphthong or consonant). Function-words were analyzed separately. Consonants were divided into voiced consonants, sonants and surds. Amounts of different types of sounds obtained from the analyzed texts are provided in Table 1.

Table 1. Amounts of sounds used for analysis

Quantity Sounds	Root	Ending	Prefix	Suffix	Function word
Vowels	169	138	40	39	54
Diphthongs	28	21	–	5*	5*
Surds	167	39	27	21	40
Voiced consonants	80	6*	–	4*	16
Sonants	201	–	11	37	29
Consonants	448	45	38	62	85

* Analysis was not performed due to small amount of data.

Ratio between sound duration in slow and normal rates: averages for defined sound groups together with 95% confidence intervals, for two speakers are given in Fig. 1. It may be noted that though the general ratio how much the rate has changed differs between speakers, a trend for both speakers exists that the ending sound length varies more than the root length. Significant differences were obtained only for the vowels of the first speaker, and the consonants of the second speaker. This illustrates how much speech characteristics are speaker-dependent: not only sound durations characterize a speaker, but also more complex features, e.g. how duration of different sound classes vary with varying rate.

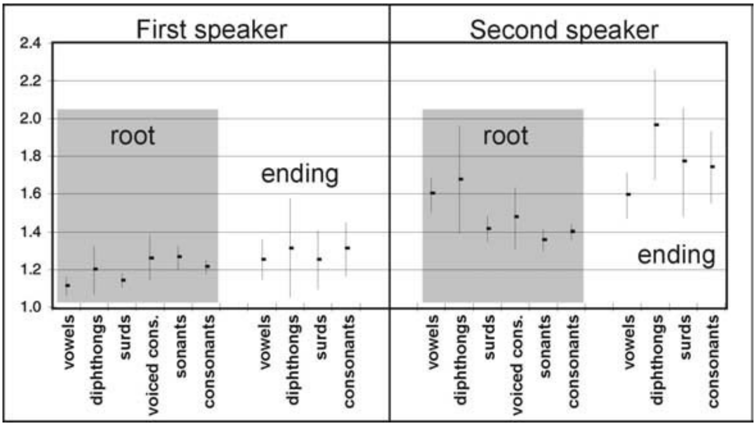


Figure 1. Ratio between sound length in slow and normal speech rate with 95% confidence intervals, root and ending, two speakers

Further average results for three speakers are presented, in an attempt to more clearly show regularities in duration change, this time in a ‘speaker-independent’ manner. In Fig. 2 the average proportions between slow and normal rate sound lengths are presented. It may be observed, that root sound lengths vary less than the ones in the ending, and prefix sounds vary a fraction less than the ones in the suffix. Vowels in all parts of the word shorten approximately equally, around 1.3 – 1.4 times. The diphthongs shorten the most in the suffix and in the ending (~1.6 times), and the least in the root (~1.4 times). Voiced consonants duration varies most from all sound groups: in root shortens 1.3 times, in short words – just 1.15 times, and in the ending – 1.6 times. The surds and sonants shorten similarly from approx. 1.3 times (in the prefix) to 1.4-1.5 times (in the ending).

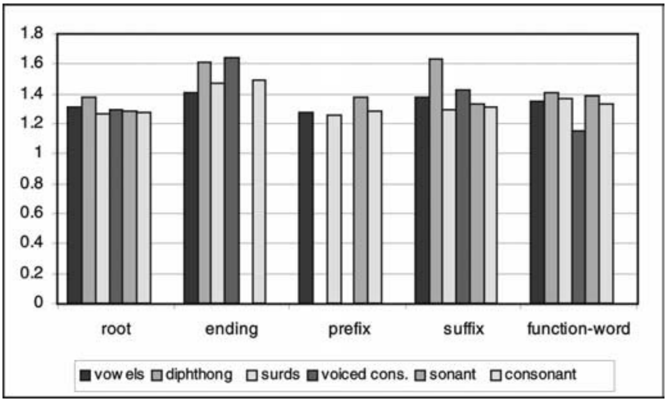


Figure 2. Ratio between sound length in slow and normal speech rate, average for three speakers

Changing rate of speech from normal to fast, general rules observed for slow/normal ratio hold: sounds in root and prefix shorten less than in the suffix and the ending. Only behavior of diphthongs changes to some extent: they vary less in comparison to other sounds than at the transition from slow to normal rate.

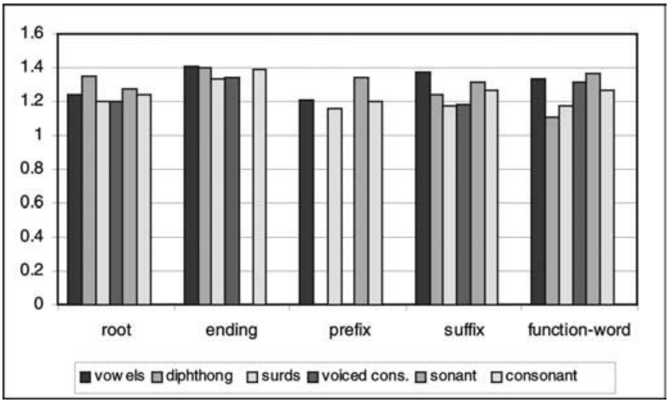


Figure 3. Ratio between sound length in normal and fast speech rate, average for three speakers

4. Pitch analysis

Pitch was analyzed at the level of syllables. In the first part of experiments two pitch models were considered: (1) rising in the stressed syllable, and falling in the post-stress syllable, and the other one (2) rising and falling in the same stressed syllable. These two models were successfully applied for two different accentuations of stress in Swedish language (Botinis et al., 2001).

Pitch was calculated with PRAAT program every 0.01 s, in the regime "very accurate" using Gaussian window with minimum and maximum pitch values 75 Hz and 500 Hz respectively, silence threshold 0.03, voicing threshold 0.45, octave cost 0.01, octave jump cost 0.35, voiced/unvoiced cost 0.14. A total amount of 169 stressed and post-stress syllables were analyzed throughout the two texts.

33 % of all stressed syllables behaved according to the first model of rising in the stressed syllable, and falling in the post-stressed one, and in 31% of syllables pitch was rising and falling in the same stressed syllable (two times faster). Significant correspondence between the two models and types of Lithuanian accentuations was not observed. The rest 36% of cases could not be attributed to either of the models. These might be due to intonation component at the phrase or sentence level which was not investigated here, or otherwise different pitch behavior from the hypothesized ones could be specific to Lithuanian language.

To summarize information about the scope of forms that the pitch behavior takes, in the second part of the experiment piece-wise linear pitch model was introduced. Syllables were divided either into two or three parts, or not divided at all, and three possible behaviors of each part were considered: rising, constant or falling. The pitch was considered constant if it was varying no more than 5 Hz throughout the corresponding time interval. Let us introduce the following notation: rising pitch – "0", constant pitch – "1", and falling pitch – "2". Then e.g. for a three piece division of a syllable notation "002" will denote rising-rising-falling tone.

A distinctive pattern is clearly observable if stressed syllables are divided into three parts and the post-stressed ones into two (Fig. 4). The pattern emergence in different models corresponds to the phonetic rule that stressed syllables in Lithuanian speech are longer then the non-stressed ones. Similar pitch pattern study was performed for Australian-English (Grigoriu 1994), where the authors also were discovering patterns, although only coarser pitch variations were taken into account there, referring to the observation that a listener can only hear a change of 30-50 Hz in a 0.1 s interval.

For Lithuanian language the prevalent models for stressed syllables were “002”, “000”, “001”, denoting rising patterns with also constant or falling behavior in the last part. Also “112” and “012” could be distinguished as more frequent, rather denoting constant-falling pattern. And complex patterns with the third falling part, like “120”, “010”, “020” were specifically infrequent.

For the post-stressed syllable falling intonations “22”, “21”, “12” were significantly prevailing, while “02” (rising-falling, second model of the first part of experiments) were also observed.

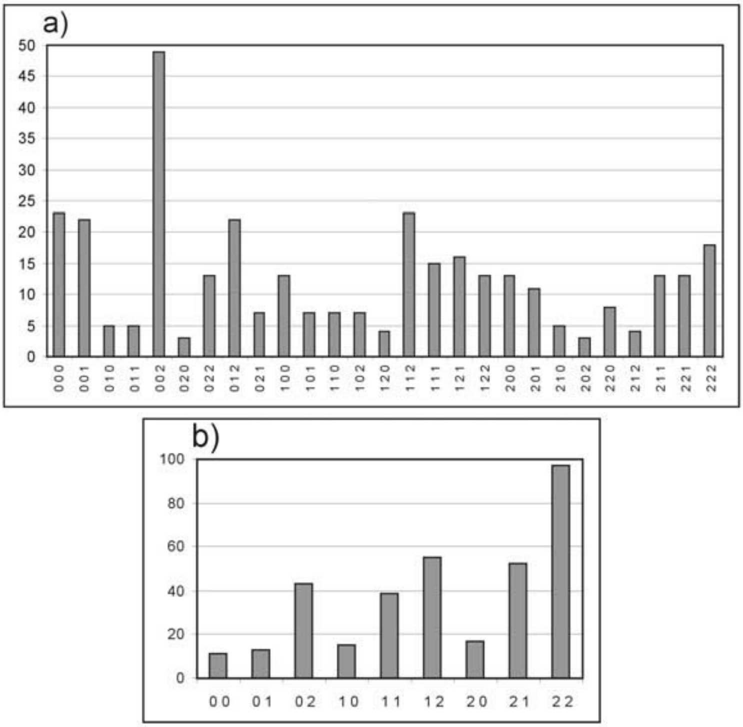


Figure 4. Stressed and post-stressed syllables’ distribution: in (a) 3-part stressed and (b) 2-part post-stressed

Taking the observed regularities into account we propose the preliminary model for Lithuanian syllable pitch generation, where for the stressed syllable pitch is rising, and beginning to fall, and falling for the post-stressed syllable (Fig. 5).

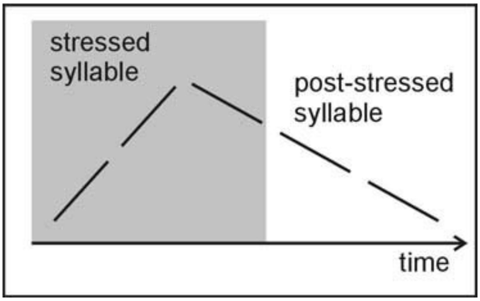


Figure 5. Pitch model for stressed – post-stressed syllable pair

5. Conclusions

A trend was observed that the word ending and suffix sound duration is more variable than that of root and prefix, with varying Lithuanian speech rate. The result may be found useful for rate normalization issue when creating sound duration models. Preliminary model for pitch in the stressed and post-stressed syllable is with three parts for stressed syllables with the profile: rising-rising-falling, and falling in the post-stressed syllable.

References

Botinis, Antonis; Granström, Björn; Möbius, Bernd 2001. Developments and paradigms in intonation research. *Speech Communication* 33. 263–296.

Chung, G.Y.; Seneff, S. 1999. A hierarchical duration model for speech recognition based on the ANGIE framework. *Speech Communication* 27. 113–134.

Grigoriu, A.; Vonwiller, J.P.; King, R.W. 1994. An automatic tone contour labeling and clasiffication algorithm. In: *Proc. ICASSP'94*, Adelaide, Australia. 181–184.

Jance, E.; Nooteboom, S.; Quene, H. 2003. Word – level intelligibility of time–compressed speech: prosodic and segmental factors. *Speech Communication* 41. 287–301.

Kazlauskienė, Asta 2000. Influence of neighboring sounds for quantitative parameters of vowels in south-west Aukštaičiai dialect (in Lithuanian). *Kalbotyra* 48–49. 63–70.

Keller, Eric 2002. Towards greater naturalness: Future directions of research in speech synthesis. In: Keller, E.; Bailly, G. et al. (eds.) *Improvements in speech synthesis.* England: John Wiley & Sons. 3–18.

Pakerys, A. 1982. Lithuanian Language Prosody (In Lithuanian). Vilnius: Mokslas.

Vaitkevičiūtė, V. 1995. Accentuation in Lithuanian speech (In Lithuanian). Vilnius.

INGRIDA RADZIUKYNIENĖ is a Master Program student at Vytautas Magnus University, Department of Informatics, Kaunas, Lithuania. Her research interests include speech analysis, and asset pricing modeling; author of three publications.
E-mail: Ingrida_Radziukyniene@fc.vdu.lt

LARGE VOCABULARY AUTOMATIC SPEECH RECOGNITION FOR RUSSIAN LANGUAGE

A.L. Ronzhin, A.A. Karpov
St. Petersburg Institute for Informatics and Automation, St. Petersburg, Russia

Abstract

The paper describes the process of development of computer model of speech recognition, which provides speaker independent input of Russian language. In contrast to English the Russian language has much more variety on word-form level and so the size of recognized vocabulary sharply increases as well as quality and speed of the processing decrease. To avoid these problems the additional level of speech representation (morphemic level) was applied. This allowed to reduce the size of vocabulary of base lexical units in several orders. The databases of various types of morphemes and automatic methods of text processing were developed on the base of the rules of Russian word-formation. We tested the developed speech recognition system for medium vocabulary only and obtained 95% accuracy of speech recognition for 2000 words. This quality is enough to continue this research, increase the amount of recognized words and realize speech recognizer for large vocabulary.

Keywords: information retrieval, speech recognition, voice interface, dialogue model, Slavonic languages

1. Introduction

Information technologies more and more penetrate into a daily life of each person. Modern technical devices develop aside intellectualizations and automation of their services. The systems of an artificial intelligence connected with pattern recognition, the analysis of images and speech especially actively are developed. Speech technologies find the increasing distribution in robotics, control systems of the equipment, means of telecommunications. The elaboration of new services and systems, which could maximally use various communication abilities of a human and, first of all, the natural speech, is one of the most perspective directions in telecommunication market. All these perspective applications are reality in the USA and Europe. In Russia some attempts to develop and introduce similar systems and services are now undertaken. The basic problems arising by development of intellectual speech systems, are connected to specificity of Russian, in particular, with the complex mechanism of word-formation.

The main aim of the basic research is the development of computer model of voice interface, which provides speaker independent input of Russian speech. For large vocabulary speech recognition with appropriated quality it is necessary to use fully all the linguistic rules and databases of Russian language. In contrast to English the Russian language has much more variety on word-form level and so the size of recognized vocabulary sharply increases as well as quality and speed of the processing decrease. Moreover usage of the syntactic constraints leads to that the errors of declensional endings cause the recognition error of the whole pronounced phrase. To avoid these problems the additional level of speech representation (morphemic level) was inserted. Owing to division of word-form into morphemes the vocabulary size of recognized lexical units is significantly decreased, since during the process of word formation the same morphemes are often used. The developed modules and databases were combined in joint hardware-software complex for automatic input and recognition of Russian speech entitled SIRIUS (SPIIRAS Interface for Recognition and Integral Understanding of Speech). The developed SIRIUS model was applied for the creation of the voice access to the electronic catalogue “Yellow pages of Saint-Petersburg".

In this paper the specifics of Russian language, which is important for automatic speech recognition, are considered in section 2. Section 3 describes the structure of the databases and algorithms of their creation. Section 4 contains the developed model for Russian speech recognition. The structure of model and main processing levels are presented there. In section 5 an experimental evaluation of the speech recognition model is presented.

2. The specifics of Russian language

To develop effective speech recognition system for Slavonic languages (and in particular for Russian) it is necessary to solve some difficulties concerning the peculiarities of these languages. These languages belong to the category of synthetic languages, which are characterized by tendency to combination (synthesizing) of the lexical morpheme (or several lexical morphemes) and one or several grammatical morphemes in one word-form. Thus during development of Slavonic speech recognition systems with large vocabulary it is required to use recognizable vocabulary much more than for English, that essentially decreases the accuracy and speed of recognition. For instance, in leading speech recognition system for English (from Microsoft, IBM, ScanSoft, etc.) the vocabulary contains 100-150 thousands of words. For Russian language the recognizable vocabulary must be increased in several orders because of existence of prefixes, suffixes, endings. The Zaliznjak’s grammatical dictionary of Russian language contains about 160 thousands of the words and owing to the especial system of signs it allows to construct all the word-forms for the words. Extracting all the dictionary entries we obtain over 3.7 millions of word-forms, and it is very large recognizable vocabulary. Moreover, the most of word-forms of the same word are differed in the endings only, but usually a human pronounces the endings of words indistinctly (not clearly) in comparison with beginning part of word. And the errors in the endings of words during speech recognition lead to the errors of phrase recognition, which appear because of non-coordination of words in the sentence.

Also other specific features of Russian language, which complicate the recognition of Russian speech should be mentioned. The order of words in Russian sentences is not defined by strict grammatical constructions (in comparison with English or German). It complicates the creation of statistical language models based on bi-grams or N-grams as well as grammars, and decrease their effectiveness.

There is diverse phonetic structure of Russian and other (non Slavonic) languages. In International phonetic alphabet SAMPA for Russian there are 44 phonemes: 38 phonemes for consonants and 6 phonemes for vowels. In American version of SAMPA for English there are 24 consonants and 17 vowels (including set of diphthong). It is clear that recognition of consonants is more difficult than vowels since they are less stable than vowels and have less duration in time.

The words in Russian language are longer in average that slows down the process of recognition since it is required to analyze longer fragments of the signal. Also the complexity of Russian speech recognition connected with the problem that Russia is multi-national country. As a result we have the set of accents and dialects that is difficult to take into account during creation of speaker-independent speech recognition system since it is required to collect huge speech databases for the training acoustical models. For training English speech recognition systems there are many speech databases (clear speech, telephone speech, etc.) including free available. But Russian speech databases are just being developed.

3. The development of the databases

Since we have introduced the morphemic level of processing so the morphemes vocabulary is needed here. The creation of morphemes databases was accomplished by published dictionaries. Most of the root morphemes (about 4000) were taken from Kuznetzova's morphemes dictionary of Russian language. Besides, the morphemes vocabulary was constantly increased and now it contains about 5000 different morphemes. The developed lexical databases, initial vocabularies and other databases required for speech recognition are described in (Ronzhin & Karpov, 2004). Text transcription and morphemes presentation modules can be marked among text processing modules.

The modules of transcription and morphemic representation construct the transcribed vocabulary of morphemes for the concrete domain. This process is started from the analysis of phrases of domain. Then the word vocabulary of domain is constructed and all words are divided into morphemes. At that new morphemes are added to the full vocabulary of morphemes. The division of word into morphemes is fulfilled by combining the different types of morphemes and the rules of morphemes order in the word.

The text with size about 50Mb was used ("Electronic Library") for language model construction. This text was preliminary processed and all words were divided into morphemes. At that the segmentation into words and sentences was saved and morphemes were marked by types. As a result of text analysis the morpheme vocabulary with size about 5000 morphemes and the statistics of all the encountered morphemes were obtained. To construct a language model for the concrete domain the analysis of text of domain is accomplished and earlier created language model constructed on existent texts was used.

One of the main modules of the system is the transcription module. The text transcription module was developed by using the following initial data: (1) the Zaliznyak's grammatical vocabulary; (2) the morphological rules; (3) the rules of stress positioning; (4) the rules of transcription. At the input of module of text transcription of subject domain the following information is entered: the set of sentences of subject domain; the words vocabulary obtained from these sentences and divided into morphemes; the word-forms vocabulary obtained from base word-forms of Russian language with marks about stressed syllable (syllables); the phonetic alphabet and the rules for automatic transcription.

As a phonetic alphabet we use the modified International phonetic alphabet SAMPA. In our variant there are 48 phonemes: 12 vowels (taking into account stressed vowels) and

36 consonants (taking into account hard consonants and soft consonants). During the transcription process the following positional changes of sounds are possible: changes of vowels in stressed position; changes of vowels in syllables before stress; changes of vowels in syllables after stress; positional changes of consonants. Moreover, for more correct transcription in some cases the information about division of words into morphemes is required.

The acoustical modeling in our system is based on Hidden Markov Models (HMM). Each base recognizable item (phoneme or triphone) is represented by some kind of HMM, for which the parameters training is performed according to training set of speech data. It is required to consider that training set of acoustical data for HMM should be large enough to take into account possible spectrum of users of a system. As acoustical models we used HMM with mixture Gaussian probability density functions (Young et al., 2000). For feature extraction we used the mel-cepstral coefficients with first and second derivatives. The phonemes are used as triphones (the phoneme in some phonetic context). For the creation of speech databases the texts (isolated sentences of subject domain) were beforehand prepared and the voices of several speakers were recorded. Based on these data the procedure of training the acoustical models (HMM of triphones) was performed.

Thus the described process of database preparation for the concrete domain and training the acoustical models allows constructing the required applications of Russian speech recognition. The developed methods and the modules for Russian speech recognition as well as speech signal processing are described in the next section in detail.

4. The continuous speech recognition model

The beforehand prepared databases for the concrete domain, namely: the transcribed morphemes vocabulary of domain; the morphemic language model of domain and the acoustical models of phonemes of domain are further used for continuous speech recognition. However, if during the preparation of database we analyze phrases by dividing them to phonemes then now the opposite process is realized: most probable chains of phonemes are composed and morphemes, words and phrases are consequently synthesized. This process is shown in Figure 1. Below each level of the speech signal processing is considered.

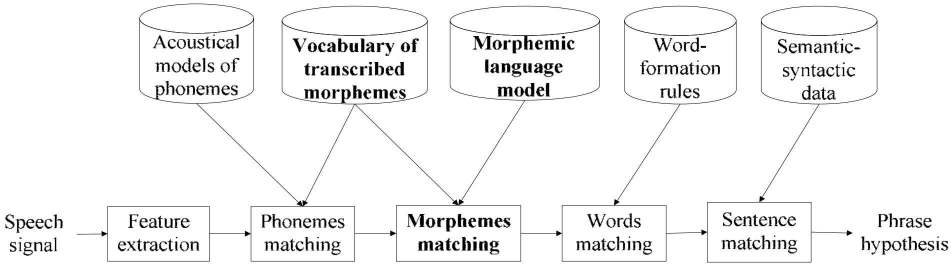


Figure 1: The matching the phrase hypothesis from speech signal

The speech signal captured through a microphone firstly passes the stage of parametric representation, where starting and ending pauses in the signal are eliminated and speech part is encoded into the sequence of feature vectors. The parameterized speech signal follows to the module of phoneme recognition. The recognition of phoneme (triphones was used in our last version) and the formation of morphemes are based on methods of HMM and Viterbi algorithm. In contrast to existent analogues our model uses the morpheme level instead of the word one. Due to this change the recognition of lexical units was significantly accelerated. At that in comparison with words-based recognition the accuracy of morpheme recognition was slightly degreased but due to the following levels of processing the accuracy of phrase recognition was not changed almost.

After the phoneme recognition and matching the obtained set of hypotheses of most probable sequences is used for synthesis of word sequences. The process of word synthesis from different types of morphemes is realized by oriented graph, which consists of the starting, ending nodes as well as nodes presented the different types of morphemes. The arcs denote possible transitions. In future this model would be stochastic but now the maximal number of transitions from node to node is given inflexibly. Every phrase hypothesis presented by morpheme sequence generates other hypotheses presented by sequence of words.

The last processing level is the sentence matching. The most errors of word recognition in the previous levels are connected with the errors in recognition of ending morphemes and prefixes. Thus, for instance, the recognized word-form and said word-form can belong to the same word, but have diverse endings. To solve this problem now we use the approach based on the dynamic warping of sentences. The objective of the approach is a search of the optimal matching for two sentences (recognized phrase and reference phrase), which are presented as sequences of characters. A measure, which allows to define difference degree between characters, is logical comparison of characters (0 - if two compared characters are equal, and 1 - otherwise). Then the optimal path is searched. The usual recurrent equations are used here (Kosarev et al., 2002). The comparison of recognized phrase and all reference phrases (for instance, rubric titles) gives the optimal reference phrase, which corresponds to the recognized phrase according to the criteria of minimum of deviation. This phrase is the result of speech recognition. Further the investigation of semantic-pragmatic processing is being carried out in the group and in near future the obtained results will be introduced into the model of Russian speech recognition. Additionally to the SIRIUS engine the required software and databases were developed: (1) the databases of different types of morphemes; (2) the software modules for automatic word formation; (3) the module for automatic text transcribing; (4) the module for morphological parsing and accumulating the statistics of existent pairs of morphemes by text.

5. Experiments

The task of voice control by rubricator of the electronic catalogue “Yellow Pages of Saint-Petersburg” was selected for debugging of the model of speech recognition. This catalogue is located on web site http://www.yell.ru. It is full telephone directory of Saint-Petersburg. The catalogue contains list of all the organizations and companies with reference on address, phones and types of activity. The size of word vocabulary for thematic rubrics was 1850 words that was used for preliminary experiments. The 635 phrases (which contain 2574 words) were recorded by 5 speakers in the office environment and the accuracy of speech recognition was over 95%. Moreover, the comparison of performance of morphemes-based recognizer with words-based recognizer has shown that the accuracy of phrase recognition is not changed almost and the developed system works in 1,7 times faster.

6. Conclusion

The conducted research is directed to the investigation of peculiarities of Russian speech and the creation of large vocabulary speech recognition as well as application this basic research to the development of intellectual services for information retrieval. The voice access to Yellow pages is being developed and tested now. In this task the size of vocabulary was significantly decreased due to introduction of the morphemes level of speech representation. The achieved quality of speech recognition is enough to continue this research and increase the amount of recognized words and realize large vocabulary speech recognizer for Russian language.

Acknowledgement

This work is supported by Russian Science Support Foundation, SIMILAR NoE FP6:IST-2002-507609 and INTAS Project № 04-77-7404.

References

Electronic Library of Maxim Moshkov, from http://lib.ru/

Kosarev, Y. A.; Lee, I. V.; Ronzhin, A. L.; Skidanov, E. A.; Savage, J. 2002. Survey of the approaches to speech and text understanding. SPIIRAS Proceedings, Issue 1, Vol. 2. St. Petersburg: “Anatolya”, 157–195.

Ronzhin, A. L.; Karpov, A. A. 2004. Implementation of morphemic analysis to Russian speech recognition. Proceedings of International Conference SPECOM’2004. St. Petersburg: “Anatolya”, 291–296.

Young, S.; et al. 2000. The HTK Book v3.0. Cambridge University Engineering Department.

ANDREY RONZHIN is head of the Speech Informatics Group of SPIIRAS. Scientific secretary of Dissertation Council of SPIIRAS. Chair of Organizing committee and member of scientific committee of the 9th International Conference “Speech and Computer” SPECOM’2004. He received PhD at SPIIRAS, dealing with automatic speech recognition. The laureate “Distinguished PhD of the Russian Academy of Sciences 2004 and 2005”, Russian Science Support Foundation. "Youth prize of Saint-Petersburg 2004" was rewarded by Public council of Saint-Petersburg. The on-going projects in framework of FP6 and INTAS programs are dedicated to the creation of the effective model of human-computer interaction. E-mail: ronzhin@iias.spb.ru.

ALEXEY KARPOV is PhD student at SPIIRAS. He graduated Saint-Petersburg State University of airspace instrumentation with honour degree. His research interests are connected with digital signal processing, Russian speech recognition and multimodal interfaces. His doctoral study focuses on acoustical models of Russian phonemes in noisy environments. He received award "Distinguished PhD student of the Russian Academy of Sciences 2005". The laureate of competition "Grant of Saint-Petersburg 2004" of ISSEP. The winner of competition of personal grants for 2004 for young scientists and specialists of Saint-Petersburg and North-West of Russia. The on-going projects in framework of FP6 and INTAS programs. E-mail: karpov@iias.spb.su.

AN EXPERIENCE OF CREATING LITHUANIAN SPEECH-ENABLED WEB APPLICATIONS

Algimantas Rudzionis*, Kastytis Ratkevicius*, Vytautas Rudzionis, Pijus Kasparaitis****
* Kaunas university of Technology, Lithuania
** Vilnius university, Lithuania

Abstract

The activities to develop Lithuanian Web pages for speech-enabled access from telephone are presented. After the integration of Lithuanian text-to-speech synthesizers "Aistis" and LtMBR to SAPI 5 (Speech Application Programming Interface), some Lithuanian speech-enabled Web pages were prepared (www.speech.itpi.ktu.lt). New version of Lithuanian text-to-speech synthesizer LtMBR is described. Two methods of SALT (Speech Application Language Tags) technology implementation to Web pages are presented. First attempts to master Microsoft Speech Server'2004 are discussed.

Keywords: SALT, text-to-speech synthesis, speech-enabled Web pages, speech recognition

1. Introduction

Speech enabled Internet services are one of the most rapidly growing fields of speech technology and it's applications. This growth creates background to develop standards for speech technology implementation over Internet. SALT and VoiceXML are of particular importance for speech technology applications.

Both Speech Application Language Tags (SALT) (SALT forum 2005) and Voice Extensible Markup Language (VoiceXML) (VoiceXML forum 2005) are markup languages. Their purpose is to enable to develop programs capable to receive and translate voice messages from web pages. Both these languages were created by industry consortiums (SALT forum and VoiceXML Forum) together with World Wide Web (W3C) consortium.

VoiceXML first appeared in 1999, as Hypertext Markup Language (HTML) branch. It originally has been developed to support phone menus and other telephony functions in applications using voice processing.

In 2002 SALT forum presented new markup language - SALT, which has the same features as VoiceXML and additional ones, such as implementation of speech technologies for more devices, such as PDA's, TabletPC's. SALT allows interactive telephony dialog forms (as VoiceXML) and multimodal dialog forms.

SALT allows add speech content, such as voice input and output, to already created web page. This technology allows read the content of a webpage using text-to-speech (TTS) engine or prerecorded files, browse by speaking voice commands, fill in internet forms by speaking, leave voice messages and create new possibilities for the customer.

Processing of SALT applications is done differently depending on the type of client platform. For example, when a user accesses speech enabled webpage from a computer, the computer will use local resources such as installed TTS engine for speech output and recognition engine for speech input. If a speech enabled webpage (or application) is accessed from PDA or telephone, all processing will be done on server side. For this purpose Microsoft developed Speech Server for its Windows 2003 server family.

2. Microsoft Speech Server

A new member of the Microsoft Windows Server System family of products, Microsoft Speech Server was created to deliver a flexible, integrated speech platform that delivers the business value of speech in a truly cost effective manner. Built on the Microsoft Windows Server 2003 operating system, Microsoft Speech Server provides high reliability and availability, increased performance and scalability, and a wide range of advanced security features.

Microsoft Speech Server also includes a set of SALT development tools that take advantage of the Microsoft Visual Studio.NET Web-programming model. Together Microsoft Speech Server and the speech development tools form a platform for building speech applications that offers low total cost of ownership, the flexibility to meet real-world business needs and the ability to interoperate with existing applications and solutions.

In addition, because Microsoft Speech Server combines Web technology with speech-processing services and telephony capabilities in a single system, developers can create applications that provide either voice-only or multimodal features. With Microsoft Speech Server, companies can deploy speech-enabled solutions that can be accessed by telephone, cell phone, Pocket PC, Tablet PC and other devices (Microsoft Speech Server 2005).

To use these technologies for Lithuanian language, one must have a Lithuanian TTS and recognition engines installed. Two Lithuanian text-to-speech synthesizers “Aistis” and LtMBR are integrated to SAPI 5 while Lithuanian speech recognizer compatible with SAPI 5 so far is under development (Rudzionis et al. 2004).

3. Lithuanian text-to-speech synthesizer LtMBR

One of the components of the speech-enabled Lithuanian Web applications is the new Lithuanian text-to-speech synthesizer LtMBR. The voice is generated by MBROLA speech synthesizer (dynamic link library "mbrola.dll" and the diphone database "lt2"). They can be freely downloaded (for non-commercial non-military use) from MBROLA project home page (http://tcts.fpms.ac.be/synthesis/mbrola/mbrcopybin.html). Besides the principles of building of the database "lt2" are published in (Kasparaitis 2005). The names of sounds, their durations and the curves of fundamental frequency are put to the input of the MBROLA synthesizer. The names of sounds are produced by automatic stressing and transcribing the text. The stressing algorithms are published in (Kasparaitis 2000) and (Kasparaitis 2001), the transcription algorithm was published in (Kasparaitis 1999) and its modified version in (Kasparaitis 2005).

The model proposed by D. Klatt was used for duration modeling (Klatt 1979). The duration D of a sound can be calculated according to the formula (1):

(1) $D = D_{minimum} + (D_{inherent} - D_{minimum}) * \Pi f_i$.

where factors f_i define both linguistic (e. g. context) and non-linguistic (e. g. speaking rate) factors. Factors f_i are found experimentally.

After some simple experiments were carried on, the inherent and minimal durations of the sounds were estimated, 7 most important factors were found and their values were calculated. The factors are as follows: vowel before vowel, vowel after vowel, vowel after voiced consonant, vowel before the group of consonants, consonant belonging to the group of consonants, consonant at the end of phrase, vowel at the end of phrase before consonant, vowel at the end of phrase. In addition to the mentioned factors on more factor was used - the speaking rate.

The curves of fundamental frequency F_0 were modeled as a superposition of phrase intonation curves and pitch accent curves (Fujisaki et al. 1998). The only difference is that another function for modeling the pitch accent curves was chosen (Dobnikar 1996). The fundamental frequency contour can be calculated according to the formulas (2-4):

$$(2) \qquad \ln F_0(t) = \ln F_b + \sum_{i=1}^{I} A_{p_i} G_p(t - T_{0i}) + \sum_{j=1}^{J} A_{a_j} G_a((t - T_{1j})/d_j)$$

$$(3) \quad G_p(t) = \begin{cases} \alpha^2 t e^{-\alpha t}, & t \geq 0, \\ 0, & t < 0, \end{cases}$$

$$(4) \quad G_a(t) = \begin{cases} 1 + \cos(\pi t), & -1 \leq t < 1, \\ 0, & t < -1, t \geq 1 \end{cases}$$

where $G_p(t)$ represents the impulse response function of the phrase control mechanism and $G_a(t)$ represents the impulse response function of the accent control mechanism. The symbols in these equation indicate: F_b – baseline value of fundamental frequency, I – number of phrase commands, J – number of accent commands, A_{pi} – magnitude of the *i*th phrase command, A_{aj} – magnitude of the *j*th accent command, T_{0i} – onset of the *i*th phrase command, T_{1j} – middle of the *j*th accent command, d_j – duration of the *j*th accent command, α – parameter for F_0 shape control (equals to 3).

The pitch of the synthesized speech can be controlled by controlling the baseline of the fundamental frequency.

The sample speech synthesizer "SampleTTSVoice" from Microsoft Speech SDK v. 5.1 was used when creating the interface of the synthesizer LtMBR, so the synthesizer LtMBR is partially compatible with SAPI 5.

4. SALT overview

There are four main elements of SALT: “listen”, “prompt”, “dtmf”, and “smex”. The other elements: “param”, “grammar”, “record”, “bind”, “value” and “content” occur only as child elements of the top-level elements (Graham 2005).

The “listen” element is used for speech recognition, for audio recording or for both. A “listen” element which is used for speech recognition contains one or more “grammar” elements, which are used to specify possible user inputs. A “listen” element which is used for audio recording contains a “record” element which is used to configure the recording process. A “listen” element used for simultaneous recognition and recording holds one or more “grammar” elements and a “record” element. In all cases, “bind” can be used to process the results obtained from recognition and/or recording.

The “prompt” element is used to specify the content of audio output. The content of prompts may inline or referenced text, which may be marked up with prosodic or other speech output information, variable values retrieved at render time from the containing document or the links to audio files. “Prompts” can be specified and played individually, and, in more complex applications, they may be managed through a model of “prompt” queuing.

The “dtmf” element is used in telephony applications to specify possible DTMF inputs and a means of dealing with the collected results and other DTMF events.

“Smex” (Simple Messaging EXtension) element communicates with the external components of the SALT platform. It can be used to implement any application control of platform functionality such as logging and telephony control.

In the mean time, the most popular software packages for integrating speech into web pages are:

- Microsoft Speech Application SDK (SASDK) with Microsoft Visual Studio.NET;
- VoiceWebSolution's “Voice Web Studio” plug-in for Macromedia Dreamweaver MX.

5. Microsoft Speech Application SDK

Microsoft Speech Application SDK (SASDK) is freely downloadable add-on and documentation package for Microsoft Visual Studio.NET 2003 for creating speech enhanced applications.

With SASDK, user can develop speech applications as: touch tone interfaces (e.g. DTMF), voice-driven menus (e.g. IVR), and multimodal interfaces (e.g. WebPages). Applications are developed using ASP.NET, adding speech to standard Web applications using SALT.

SASDK includes speech control, grammar and prompt editors, sample applications and libraries, logging and debugging tools, and speech add-ins.

ASP.NET Speech Controls are ASP.NET controls that render SALT in a speech-enabled Web application. Using property “builders” developer can set most common properties of Speech Controls in an organized, intuitive format. Using Speech Application Wizards developer can quickly configure settings for both voice-only and multimodal applications, creates a template containing prompt projects, a blank grammar file, the grammar library, and application-specific debug settings.

Speech add-ins allow browser to recognize SALT and execute speech-enabled Web applications.

6. Voice Web Studio

Voice Web Studio (VWS) is an add-in for Macromedia Dreamweaver MX that enables building speech-enabled web applications based on SALT 1.0. Integration within Dreamweaver provides the developer with a familiar user interface, source and design view and compliance with multiple operating systems. Using VWS's menu in Dreamweaver, user can quickly create and insert speech content into the code by filling in interactive forms. For easier manipulation and editing of speech elements, VWS comes with build-in visual controls for displaying speech elements in Dreamweaver's design view window (Voice Web Community, 2005).

There are four main function buttons in VWS: “SALT prompt” button is used to insert a “prompt” element into the page to play synthesized text or an audio file, “SALT listen” is used to insert a “listen” element into the page, to add speech recognition feature, “SALT create dialog” is used to create speech dialogs between human and computer and “SALT play dialog” is used to select a “dialog”, “listen” or “prompt” element to activate in a multimodal setting such as clicking a textbox, mouse over an image, using the keyboard to focus in on a selection, pressing a button and so on.

7. Examples of speech-enabled Web pages

Some demonstrations of speech–enabled Web pages were prepared using Voice Web Studio and SASDK (http://www.speech.itpi.ktu.lt): a) reading of input text by voice; b) filling of forms by voice c) virtual discotheque. Speech prompts by TTS and voice dialogues are used in these applications. User can control these multimodal applications by keyboard, mouse or by voice commands. To test this web page you need to install Windows’2000 or Windows’XP system and freely distributed SASDK. Lithuanian TTS engine is needful for TTS prompts in Lithuanian.

8. Conclusions

SALT technology allow to access the content of a webpage using text-to-speech engine or prerecorded files, browse by speaking voice commands, fill in internet forms by speaking, leave voice messages and create entirely new possibilities for the customer.

Microsoft Speech Server combines Web technology with speech-processing services and telephony capabilities in a single system.

The SASDK provides an integrated, comprehensive platform for building speech-enabled ASP.NET applications. It is more powerful tool in comparison with Voice Web Studio.

Some demonstrations of speech–enabled Lithuanian Web pages were prepared (http://www.speech.itpi.ktu.lt).

References

Dobnikar, A. 1996. Modeling segment intonation for Slovene TTS system. In: *Proc. ICSLP 96, vol. 3,1996.* Philadelphia, USA. 1864 –1867.

Fujisaki, H., Ohno S., Wang C. 1998. A command-response model for F0 contour generation in multilingual speech synthesis. In: *Proceedings of the 3rd ESCA/COCOSDA International Workshop on Speech Synthesis,1998*. Blue Mountain, Australia. 299 –304.

Graham B. Speak and listen to the Web using SALT. Retrieved February 25, 2005, from http://www.developer.com/voice/article.php/2174471.

Kasparaitis, P. 1999. Transcribing of the Lithuanian text using formal rules. *Informatica, **10**(4).* Vilnius: Institute of Mathematics and Informatics.. 367 –376.

Kasparaitis, P. 2000. Automatic stressing of the Lithuanian text on the basis of a dictionary. *Informatica, **11**(1).* Vilnius: Institute of Mathematics and Informatics. 19–40.

Kasparaitis, P. 2001 Automatic stressing of the Lithuanian nouns and adjectives on the basis of rules. *Informatica, **12**(2).* Vilnius: Institute of Mathematics and Informatics. 315–336.

Kasparaitis, P. 2005. Diphone databases for Lithuanian text-to-speech synthesis. *Informatica.* Vilnius: Institute of Mathematics and Informatics. (in print).

Klatt, D. H. 1979. Synthesis by rule of segmental durations in English sentences. In: Lindblom B., Öhman S. *Frontiers of Speech Communication Research*. New York: Academic. 287–300.

Microsoft speech server. Retrieved February 25, 2005, from http://www.microsoft.com/speech.

Rudzionis A., Ratkevicius K., Rudzionis V. Speech recognition research and some speech technologies applications in Lithuania. In: *Proc. of the first Baltic Conference HLT-2004*. Riga, Latvia. 132 –138.

SALT forum. Retrieved February 25, 2005, from http://www.saltforum.org.

Voice Web Community. Retrieved February 25, 2005, from http://www.voicewebsolutions.net.

VoiceXML forum. Retrieved February 25, 2005, from http://www.voicexml.org.

ALGIMANTAS RUDZIONIS is long time head of the speech research group in Kaunas university of technology. He received doctoral degree in 1972 (research on speech compression). He is active in speech technology area since then all the time. His research interests cover speech recognition, speech compression, text to speech synthesis as well as speech enabling system design. Since 1992 he took part in several European Union COST actions (232,249,250,278). He took part in Lithuanian state research program "Lithuanian language in informative society".Email: alrud@mmlab.ktu.lt.

KASTYTIS RATKEVICIUS is senior researcher and associated professor in Kaunas university of technology. He received his doctoral degree at the Kaunas university of technology. His research interests are in speech compression, text-to-speech synthesis, speech-enabled Web applications, introducing speech technology into robotics. E-mail: Kastytis.Ratkevicius@ktu.lt.

VYTAUTAS RUDZIONIS is senior researcher and associated professor in the Kaunas faculty of Vilnius university, Kaunas, Lithuania. He received his doctoral degree at the Kaunas university of technology. His primary research interests are in the speech recognition, particularly phoneme based speech recognition, problems. Other interests are in the signal processing, speech compression and related areas. Particular interest author pays to the speech technology implementation and design of speech technology enabled systems. E-mail: vyrud@mmlab.ktu.lt.

PIJUS KASPARAITIS is a lecturer in Mathematics and informatics faculty of Vilnius university. He received his doctoral degree from Vilnius university in 2001 for his work in Lithuanian text-to-speech synthesis. He is principal developer of best up to date Lithuanian speech synthesizer. His research interests are mainly focused in speech synthesis area. P. Kasparaitis permanently works on Lithuanian text-to-speech synthesizer improvement. E-mail: pkasparaitis@yahoo.com.

A SOFTWARE TOOL FOR THE ESTONIAN DIALOGUE CORPUS

Margus Treumuth
University of Tartu (Estonia)

Abstract

The article gives an overview of Estonian Dialogue Corpus (EDiC) and describes the software tool that was created for the corpus.

The tool provides the corpus researchers with corpus queries, automated XML generation, statistical reports and dialogue transformations. It is primarily used by researchers working with EDiC but also by students of linguistics.

At the end of the article some technical aspects of the system are shortly discussed and a brief overview of other similar tools is given. The work will be continued by implementing a software tool that uses statistical learning methods for recognizing dialogue acts.

Keywords: dialogue annotation, corpus queries, frequency counts, n-grams

1. Estonian Dialogue Corpus

The Estonian Dialogue Corpus includes recordings of spoken conversations. There are currently 623 dialogues (114400 words) in EDiC, including calls for information (asking for phone numbers, addresses etc.), calls to travel agencies, and face-to-face conversations.

All the recordings were transliterated by using the conversational analysis transcription (Gerassimenko et al. 2004). Conversation analysts have developed a sign system for the transcription of talk using conventions that capture those features of talk that are important for analysis.

A dialogue act coding scheme has been developed for annotating the Estonian dialogue corpus (Hennoste, Rääbis 2004). Dialogue acts are used to describe the function of each utterance with respect to the overall progress of the interaction. The recognition of dialogue acts plays an essential role in dialogue management or in human dialogue understanding (see Example 1).

Example 1. A sample dialogue transcription

```
A: tere: (0.2) kas sa aitaksid 'mind | DIF: REQUEST |
        hello, could you help me
B: ei ma=ei=saa |DIS: REFUSAL |
        no, I can't
```

A brief explanation of the transcription notation is hereby given. Numbers in round brackets measure pauses in seconds (in this case, 2 tenths of a second). Colons show degrees of elongation of the prior sound. Equals signs mark the immediate latching of successive talk, whether of one or more speakers, with no interval. Dialogue acts are between 'pipe' (|) marks. Dialogue act names are originally in Estonian. An act token consists of two parts separated by a colon. The first two letters are an abbreviation of the act group name (e.g DI = directive, RI = ritual, QU = question). The third letter is only used for adjacency pair acts: the first (F) or the second (S) part of an adjacency pair act. The second part of a token is full name of an act.

2. Software tool for Estonian Dialogue Corpus

The software tool that was created for EDiC to study conversational interactions provides the corpus researchers with the following functionalities:

corpus queries
automated XML generation
statistical reports
dialogue transformations

The software tool consists of a database (dialogue corpus) and four essential modules (see Figure 1). These modules can be used for a single dialogue or whole dialogue corpus.

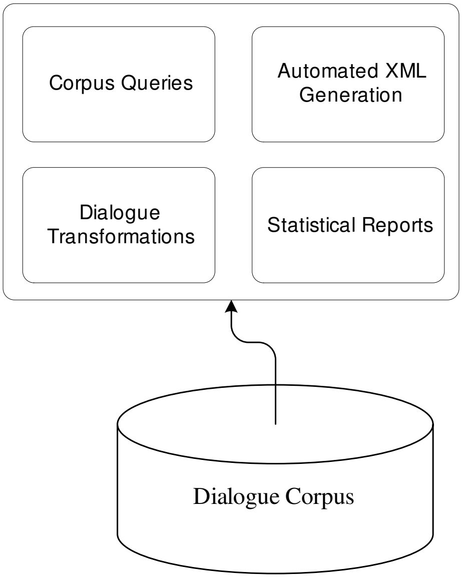
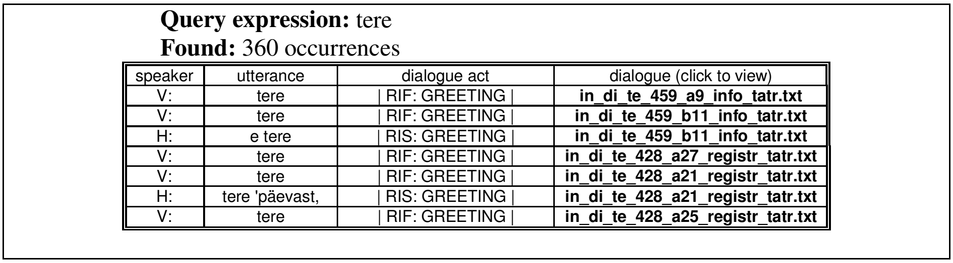


Figure 1. Functional architecture of the system

2.1. Corpus queries

The workbench supports querying the annotated corpus. Users can search the corpus for specific words or dialogue acts, according to any combination of constraints from both the transcribed dialogue text and dialogue acts annotations. The result of a query is a HTML table (see Figure 2) which can be printed or saved. For the purpose of this article the figure above and also the forthcoming figures show the user-interface in English. The actual user-interface is in Estonian.



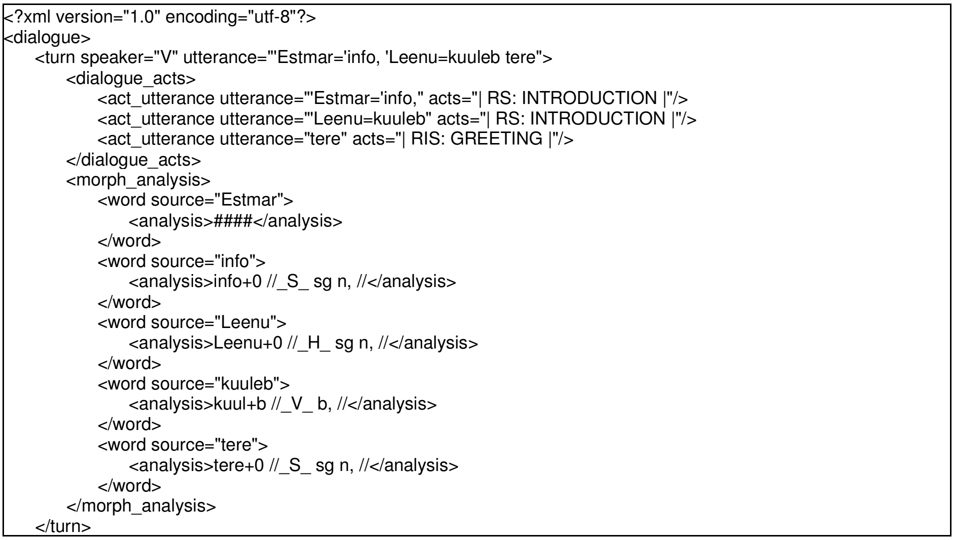
Query expression: tere
Found: 360 occurrences

speaker	utterance	dialogue act	dialogue (click to view)
V:	tere	| RIF: GREETING |	**in_di_te_459_a9_info_tatr.txt**
V:	tere	| RIF: GREETING |	**in_di_te_459_b11_info_tatr.txt**
H:	e tere	| RIS: GREETING |	**in_di_te_459_b11_info_tatr.txt**
V:	tere	| RIF: GREETING |	**in_di_te_428_a27_registr_tatr.txt**
V:	tere	| RIF: GREETING |	**in_di_te_428_a21_registr_tatr.txt**
H:	tere 'päevast,	| RIS: GREETING |	**in_di_te_428_a21_registr_tatr.txt**
V:	tere	| RIF: GREETING |	**in_di_te_428_a25_registr_tatr.txt**

Figure 2. Results of a query from the corpus

2.2. Automated XML generation

The need for generating XML arose as the dialogues were annotated on multiple levels in a plain text file. XML offers an expressive and structured way to view and edit annotations on multiple levels. XML is becoming a required standard for electronic information exchange and an XML coded corpus can be integrated with other corpus tools. During the process of XML generation, the dialogue is also annotated using the Estonian morphological analyzer (Kaalep 1998) with a disambiguator (Kaalep, Vaino 1998). The result is displayed in XML (see Figure 3).

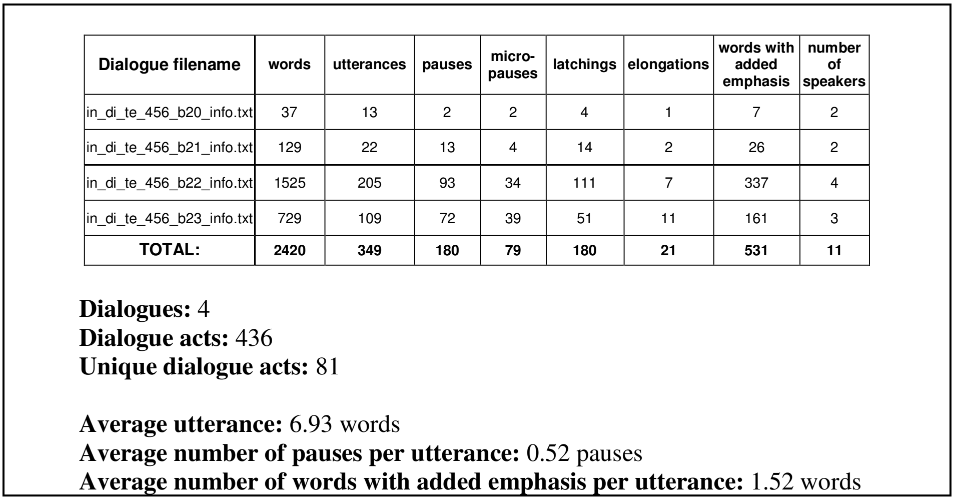


```
<?xml version="1.0" encoding="utf-8"?>
<dialogue>
    <turn speaker="V" utterance="'Estmar='info, 'Leenu=kuuleb tere">
        <dialogue_acts>
            <act_utterance utterance="'Estmar='info," acts="| RS: INTRODUCTION |"/>
            <act_utterance utterance="'Leenu=kuuleb" acts="| RS: INTRODUCTION |"/>
            <act_utterance utterance="tere" acts="| RIS: GREETING |"/>
        </dialogue_acts>
        <morph_analysis>
            <word source="Estmar">
                <analysis>####</analysis>
            </word>
            <word source="info">
                <analysis>info+0 //_S_ sg n, //</analysis>
            </word>
            <word source="Leenu">
                <analysis>Leenu+0 //_H_ sg n, //</analysis>
            </word>
            <word source="kuuleb">
                <analysis>kuul+b //_V_ b, //</analysis>
            </word>
            <word source="tere">
                <analysis>tere+0 //_S_ sg n, //</analysis>
            </word>
        </morph_analysis>
    </turn>
```

Figure 3. Result of the XML generation of a dialogue

2.3. Statistical reports

Statistical reports can be viewed on screen or printed to the printer. They can also be sent to disk in a number of formats including Word, Excel, HTML. Statistical reports can be generated for an entire dialogue corpus or any subset (see Figure 4).



Dialogue filename	words	utterances	pauses	micro-pauses	latchings	elongations	words with added emphasis	number of speakers
in_di_te_456_b20_info.txt	37	13	2	2	4	1	7	2
in_di_te_456_b21_info.txt	129	22	13	4	14	2	26	2
in_di_te_456_b22_info.txt	1525	205	93	34	111	7	337	4
in_di_te_456_b23_info.txt	729	109	72	39	51	11	161	3
TOTAL:	**2420**	**349**	**180**	**79**	**180**	**21**	**531**	**11**

Dialogues: 4
Dialogue acts: 436
Unique dialogue acts: 81

Average utterance: 6.93 words
Average number of pauses per utterance: 0.52 pauses
Average number of words with added emphasis per utterance: 1.52 words

Figure 4. Statistical report for a subset of the corpus

The software also supports calculating the frequency counts of successive dialogue acts (see Table 1), i.e.: two consecutive utterances, with different speakers producing each utterance.

This also helps to spot adjacency pairs like question/answer, offer/acceptance or refusal, invitation/acceptance or decline.

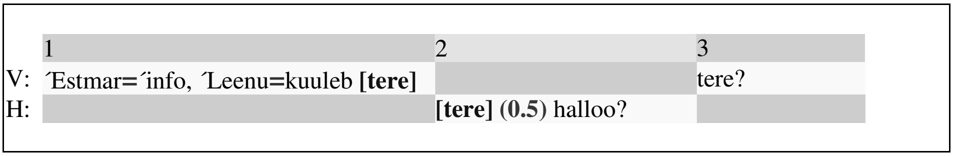
Table 1. Frequency counts of successive dialogue acts

1.	| QUS: GIVING INFORMATION | ---> | RE: CONTINUER: NEUTRAL |	98
2.	| RIF: GREETING | ---> | RIS: GREETING |	86
3.	| RE: CONTINUER: NEUTRAL | ---> | QUS: GIVING INFORMATION |	83
4.	| RIF: THANK | ---> | RIF: WELCOME |	82

There are also some syntax-highlighted reports of a dialogue that show the dialogue objects in colors to spot common transcription errors and some reports to help determine inter-coder consistency, for example listings of utterances annotated with a specific dialogue act tag.

2.4. Transformations

The tool offers some transformations on dialogues. First, it is possible to clear the annotation and output plain text, which can be used as input to the morphological analyzer. The other transformation is a graphical representation of dialogue on a timeline (see Figure 5). Visually represented dialogues also help to spot transcription errors.



	1	2	3
V:	´Estmar=´info, ´Leenu=kuuleb **[tere]**		tere?
H:		**[tere]** (**0.5**) halloo?	

Figure 5. Graphical representation of dialogue on a timeline

2.5. Implementation effort

The system was developed in PHP (PHP: Documentation) and CSS (Cascading Style Sheets Documentation) using MySQL (MySQL Documentation) as the database and Apache (Apache HTTP Server Documentation) as the web server.

The application is web enabled as the web interface offers several advantages. It is easy for the developer to add/modify functionality. It is also comfortable for users. The users do not have to install anything; there are no specific requirements for software and hardware, they just need a web browser.

2.6. Related work

There are other similar approaches where workbenches are developed to assist in corpus work.

The MATE project (MATE Multilevel Annotation, Tools Engineering) aims to facilitate the re-use of language resources by addressing the problems of creating, acquiring and maintaining spoken language corpora. MATE has developed a standard framework for the annotation of spoken dialogue corpora at multiple levels, including prosody, (morpho-) syntax, co-reference, dialogue acts, and communication problems, as well as the interaction among the levels. MATE proposes state-of-the-art best practice coding schemes for its annotation levels and has completed a workbench, i.e. a set of integrated tools, in support of the annotation framework and the best practice schemes.

The CHILDES system (CHILDES – Child Language Data Exchange System) provides tools for studying conversational interactions. These tools include a database of transcripts, programs for computer analysis of transcripts, methods for linguistic coding, and systems for linking transcripts to digitized audio and video. CLAN (Computerized Language Analysis) is a program that is designed specifically to analyze data transcribed in the format of the CHILDES. CLAN allows to perform a large number of automatic analyses on transcript data. The analyses include frequency counts, word searches, co-occurrence analyses, Mean Length Utterance (MLU) counts, interactional analyses, text changes, and morphosyntactic analysis.

3. Conclusion

There was a need for a software tool to help researchers working with the Estonian Dialogue Corpus. This led to the development of a workbench to fulfill the requirements of the researchers. Our next goal is to implement software that uses statistical learning methods for recognizing dialogue acts.

Acknowledgement

The work was supported by the Estonian Science Foundation (Grant No 5685).

References

Apache HTTP Server Documentation. Retrieved October 15, 2003, from http://httpd.apache.org/docs-project/.

Cascading Style Sheets Documentation. Retrieved December 12, 2003, from http://www.w3.org/Style/CSS/.

CHILDES Child Language Data Exchange System. Retrieved January 20, 2004, from http://childes.psy.cmu.edu/.

Gerassimenko, Olga; Hennoste, Tiit; Koit, Mare; Rääbis, Andriela; Strandson, Krista; Valdisoo, Maret; Vutt, Evely. 2004. Annotated dialogue corpus as a language resource: an experience of building the Estonian dialogue corpus. In: The first Baltic conference "Human language technologies. The Baltic perspective". Commission of the official language at the chancellery of the president of Latvia. Riga. 150–155.

Hennoste, Tiit; Rääbis, Andriela. 2004. Dialoogiaktid eesti infodialoogides. Tartu: Tartu Ülikooli Kirjastus.

Kaalep, Heiki-Jaan. 1998. Tekstikorpuse abil loodud eesti keele morfoloogiaanalüsaator. In: Keel ja Kirjandus 1. 22–29.

Kaalep, Heiki-Jaan; Vaino, Tarmo. 1998. Kas vale meetodiga õiged tulemused? Statistikale tuginev eesti keele morfoloogiline ühestamine. In: Keel ja Kirjandus 1. 30–38.

MATE Multilevel Annotation, Tools Engineering. Retrieved February 11, 2004, from http://mate.nis.sdu.dk/.

MySQL Documentation. Retrieved November 1, 2003, from http://dev.mysql.com/doc/.

PHP: Documentation. Retrieved November 11, 2003, from http://www.php.net/docs.php.

MARGUS TREUMUTH is a PhD student at the University of Tartu. He received his MSc (Computer Science) at the University of Tartu, dealing with dialogue systems interacting with a user in Estonian. His doctoral study focuses also on dialogue systems. E-mail: treumuth@ut.ee.

EUROTERMBANK TERMINOLOGY DATABASE AND COOPERATION NETWORK

Andrejs Vasiļjevs, Raivis Skadiņš
Tilde (Latvia)

Abstract

The EU eContent program project EuroTermBank focuses on harmonisation and consolidation of terminology work in new EU member states, transferring experience from other European Union terminology networks and accumulating competencies and efforts of the accessed countries. The project will result in a centralized online terminology bank for languages of new EU member countries interlinked to other terminology banks and resources. Although EuroTermBank is addressed directly towards Estonia, Hungary, Latvia, Lithuania, and Poland, the project is open to other new EU member states and interested countries and organizations outside EU.

Exchange of terminology data with existing national and EU terminology databases will be also provided by establishing cooperative relationships, aligning methodologies and standards, designing and implementing data exchange mechanisms and procedures. Through harmonisation, collection and dissemination of public terminology resources, EuroTermBank will strongly facilitate enhancement of public sector information and strengthen the linguistic infrastructure in the new EU member countries.

Public terminology resources of various owners will be acquired and processed. After that, they will be integrated into one database and made publicly available on the Internet. Another approach is planned to proprietary and copyrighted resource owners. Then, according to developed business model, they will have a choice either to provide their content for inclusion into main database or implement links to their resources. User authorization mechanism will be developed to establish commercial access to these resources via subscription or processing of requests for specific particular terminology resources on a fee basis. Part of revenues will also be utilized for maintenance and actualization of the main database.

Connections with other major digital terminology content banks will be elaborated, including data exchange mechanisms, if necessary. This will ensure possibility, for example, to have single point of service for user requests, which need equivalent of certain term in one of languages, not included in database. EuroTermBank Retrieval system will firstly look for searched term in EuroTermBank database. If not found, other terminology databases will be queried and returned results provided to the user.

Keywords: terminology, EuroTermBank, database, multilingual, eContent, methodology

1. Overview of the current situation

There is a large number of different terminological resources in Baltic countries, Hungary and Poland as well as other new EU member countries. At the same time, the overall situation in terminology area is characterized by many gaps and problems.

Resources are fragmented, located in different institutions and in different format. A lot of terminology data is available only in the form of printed dictionaries and bulletins or stored in card files. The transformation from centralised terminology development during Soviet time with the focus on Russian language to requirements of market economies is still not completed. It has lead to lack of coordination between institutions involved in terminology development, inconsistency and poor quality of terminology data, insufficient mechanisms for dissemination of new terminology.

Currently there are several important multilingual terminology resources for languages of "old" EU countries – Eurodicautom, which holds terminology of European Commission in eleven languages, TIS (Council of Ministers), Euterpe (European Parliament) and IATE project for single database for EU institutions. Only very small part of terminology entries in these databases cover languages of new EU member countries. These resources include mostly official EU terminology, leaving aside many other areas and a lot of public/private terminology resources are not networked and are difficult to access and use for wider public.

Differences in methodological approach, structuring and formatting in creation of terminology data on institutional and national level are an important terminology quality issue. There are different terminology development procedures in different institutions and countries. In some countries like in Latvia (Skujiņa 2001) there are official terminology institutions on a national level with inadequate capacity, while others have a variety of poorly coordinated public and private terminological organizations, sometimes creating parallel inconsistent versions of the same terms.

Terminology development suffers from general fragmentation of creation and distribution mechanisms on the institutional, sector/industry and national levels. Term banks tend to be small in size, mostly highly specialised, difficult to access. These difficulties are amplified by considerations of confidentiality, institutional restrictions and legal uncertainty about copyright status of certain terminology resources.

Rapid development and dissemination of new terms is especially important for smaller languages. There are several initiatives to create national terminological databases. Project partners in Latvia – Terminology Commission of Academy of Science and Tilde have participated in an initiative creating an online database termnet.lv. However, these initiatives are limited due to lack of resources and include only national terminology (Vasiļjevs, Skadiņš 2004).

As we face continuous growth of Internet penetration, centralization of access to terminological resources becomes crucial.

2. Main barriers

There are several barriers on the way to establishing a comprehensive and rich public terminology marketplace:

- A large number of creators and owners of terminology resources with weak coordination and cooperation. Traditional terminological content owners are facing difficulties using IT technologies or moving to e-business, because a large part of content requires significant investments to make it available in digital form.
- Underdevelopment of institutions, insufficient processes for standardizing and regulating terminology development in each particular country and whole regions or groups of countries.

- Low awareness of the general public about availability and access mechanisms of terminology resources. In the situation of fast-growing economical, social and political activities in Europe, ease of access to quality terminological resources becomes one of the development drivers.
- Low awareness of terminology institutions in the new EU member countries about European and EU terminology networks and databases.
- In some cases terminological resources are copyrighted. Because digital content is often easy to copy and requires use of sophisticated protection methods, owners are not publishing their content online or on digital media.
- The quality of terminology is particularly low in languages other than the term source language. In many cases – including that of a number of international bodies – there is a tendency merely to translate terms when adding equivalents. In addition, there is often a lack of conceptual organisation and of in-depth classification work with regard to sub-domains. As a result, a high percentage of irrelevant terms may be included.

3. EuroTermBank project objectives

To address previously described weaknesses and problems EuroTermBank project „Collection of Pan-European Terminology Resources through Cooperation of Terminology Institutions" or shortly EuroTermBank has started. This is eContent programm project supported by European Commission and implemented by 8 consortium partners from 7 EU countries: The Institute for Information Management (Germany), Centre for Language Technology at University of Copenhagen (Denmark), company Tilde (Latvia), University of Tartu (Estonia), Institute of Lithuanian Language (Lithuania), Terminology Commission of Latvian Academy of Science (Latvia), company MorphoLogic (Hungary), the Information Processing Centre (Poland).

Project started from the beginning of 2005 and will last for 3 years. The EuroTermBank project focuses on the following major objectives:

- Development of methodology for harmonisation of terminology processes in new EU member countries and for ensuring compatibility of terminological resources for data interchange and resource sharing;
- Creation of a network of terminology-related institutions and organizations (creators and holders of terminology resources) on both national and multinational levels to facilitate institutional cooperation and harmonisation, consolidation and dissemination of terminological resources;
- Design, development and implementation of a web-based terminology data bank to provide easy access to centralised terminology resources;
- Consolidation of terminology content from different sources and owners for creation of national terminological databases and further integration into the EuroTermBank database or their interlinking;
- Achieving sustainability of the project results.

4. Consolidation of terminology content

In brief, the goal of the EuroTermBank project is to integrate all available terminology resources (not only from project partner countries) into the central EuroTermBank database or interlink them via EuroTermBank as a central gateway and single point of service.

In general, terminology content is available in two forms – electronic and hardcopy (i.e. card files). Regardless of type, the content will require additional processing to bring them in a form ready for integration into EuroTermBank.

The methodology developed by EuroTermBank project experts will serve as the basis for content processing. The contents will pass several stages before integration into the database – prioritization, modification, and digitalization (for non-digital format).

The expected outcome is a reliable multilingual terminology resource, networked with other existing national and international resources available for users over the global network.

The project is oriented to standardization and unification of approaches, improvement of term quality and availability via term processing, digitizing and networking of resource owners and regulating bodies will have a great impact on awareness and dissemination of terminology resources on a new quality level.

5. Project approach

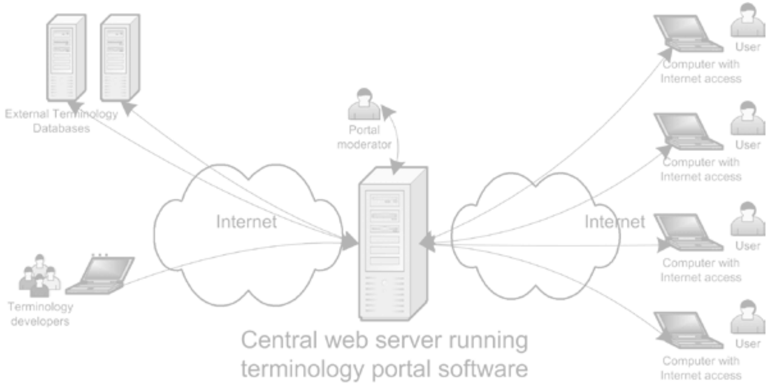
Project will establish an innovative approach to terminology dissemination and availability. At first, owners and creators of terminology resources will be identified. In this respect, activities will be focused on the participating new EU member states. In addition, development of recommendations for common standards and policies in the region will be performed on the basis of international experience and best practices.

The project will be also innovative due to the fact that access to terminology resources will be organized on two different levels:

- **Local –** Public terminology resources of various owners will be acquired and processed, integrated into one database and made publicly available on the Internet. Another approach is planned to proprietary and copyrighted resource owners. Then, according to a developed business model, owners will have a choice either to provide their content for inclusion into the main database or implement links to their resources. User authorization mechanism will be developed to establish commercial access to these resources either via subscription or by processing single requests for particular terminology resources on a fee basis. Part of the revenues will also be utilized for maintenance and updating of the main database.
- **Remote –** Connections with other major digital terminology content banks will be arranged, including data exchange mechanisms, if necessary. This will provide a single point of service for user requests, which need the equivalent of a certain term in one of the languages not included in the database. EuroTermBank retrieval system at first will look for the searched term in the EuroTermBank database. If not found, other terminology databases will be queried and results returned to the user. Search protocol will be developed in line with ISO standard requirements and will be oriented to search in all freely available databases. If a particular database structure will not allow the use of this protocol directly, additional translation mechanisms that enable the search option will be implemented.

6. Web-system features and architecture

Users will be able to query the term database in many ways. They will be able to query the full database containing terms from all fields or to choose one field and search only that particular database. They will be able to search in any language presented in the database and even search terms in all languages.


External Terminology Databases
Portal moderator
Internet
Internet
User
Computer with Internet access
User
Computer with Internet access
User
Computer with Internet access
User
Computer with Internet access
Terminology developers
Central web server running terminology portal software

Additional important search feature will be that if the user is requesting a term in a language or domain, not present in the database, this request will be automatically forwarded to another terminology resource, which contains requested information.

Support of inflectional forms while searching is very important for the languages of Baltic countries. The database will have integrated morphology, which enables the user to find terms even if they are not in the base form (Vasiļjevs et al. 2005).

The EuroTermBank will be designed for easy administrator customization. It'll provide different options for different user groups. There will be information and features accessible to everybody, as well as ones available only for authorized users. The public part will contain term bases and provide search facilities. It also will contain various public notices and documents in read-only mode. General discussions on terms and terminology will be also public.

Authorized users will have access to non-public parts of portal made available to the particular user group. Authorization principle will be used also to support commercial services of the EuroTermBank, i.e. subscription services. The administrator will be able to define access rights for each user group for each section of the portal.

Technical solution of the EuroTermBank will be an integrated platform where terms are developed, managed and published on the global network. Such platform architecture will solve many issues in terminology accessibility and development.

The system will have Multi-Tier architecture with separate layers for database, business logic and representation. XML will be used as data exchange format in the system. Standard representation and scripting technologies will be used and proprietary technologies avoided to provide access with all popular Internet browsers. Industry standard approaches for Web application development will be used in development. Data (terms, definitions and other) will be stored in SQL database. Special attention will be paid to system security.

Data exchange mechanisms will be developed for enabling of term import/export/exchange with other terminology databases and portals. Results of project activities in unification and standardization fields will form reliable and strong basis for term exchange technologies.

References

Skadiņa I. 2003. Electronic Dictionaries and Multilingual Information Society. In: *Terminology and Technology Transfer in the Multilingual Information Society*. Termnet Publisher. 140–146.

Skujiņa V. 1993. Latviešu terminoloģijas izstrādes principi. Riga: Zinātne.

Skujiņa V. 2001. Latviešu valodas terminoloģijas attīstība nacionālo un internacionālo tendenču mijiedarbībā. In: *Baltu filoloģija, X*. – Riga, LU, 21–30.

Vasiļjevs A., Skadiņš R. 2004. Multilingual Terminology Portal – termini.letonika.lv. In: Proceedings of First Baltic Conference „Human Language Technologies – the Baltic Perspective", Riga.

Vasiļjevs, Andrejs; Ķikāne, Jana; Skadiņš, Raivis 2005. Valodas tehnoloģiju izstrāde Baltijas valodām plaša izmantojuma lietojumprogrammās. In: *Latviešu valoda – robežu paplašināšana*. Valsts valodas komisija, Rīga.

RAIVIS SKADIŅŠ graduated from the University of Latvia in 1994. He is working at software company Tilde since 1995. Raivis Skadiņš has worked on development of automated English-Latvian-English dictionary, developed Latvian spellchecker, thesaurus and hyphenator. Since 1998 he is head of Language & Reference System group in Tilde. In 2001 Raivis Skadiņš received Master's Degree in Computer Science. His master thesis "Object-oriented analysis and Universal Networking Language (UNL)" investigates UNL and object-oriented analysis and studies formal means they offer to describe the real world.

ANDREJS VASIĻJEVS graduated from the University of Latvia in 1992. in 1996 he received Master's Degree in Computer Science. In 1991 Andrejs Vasiļjevs was one of the founders of Baltic software company Tilde. He is director of software development and responsible for products and services developed at Tilde. Andrejs Vasiļjevs is member of the board of Latvian Information Technology and Telecommunications Association, Soros Foundation Latvia. He is member of Commission of Official Language responsible for development of HLT.

PERFORMANCE OF *F0TOOL* - A NEW SPEECH ANALYSIS SOFTWARE FOR ANALYZING LARGE SPEECH DATA SETS

Eero Väyrynen, *Heikki Keränen, Tapio Seppänen, **Juhani Toivanen
MediaTeam, University of Oulu, Finland; *Department of English, University of Oulu, Finland; **Academy of Finland

Abstract

In this paper, the performance level of F0Tool is described. F0Tool is a speech analysis software implemented in the MATLAB language; the software is a cepstrum based voiced/unvoiced segmentation and time domain F0 extraction algorithm using waveform-matching. The performance of F0Tool was tested with a data set of 40 randomly selected sentences from radio conversations involving Finnish fighter pilots. The results of the reliability test indicate that F0Tool is a reliable software for the automatic prosodic analysis of large quanta of speech data (e.g. emotional speech); it can also be used on speech data produced in very demanding conditions.

Keywords: speech analysis, F0Tool, prosodic analysis, reliability test

1. Introduction

The scientific study of the vocal expression of emotion is now reaching a stage where the focus is on relevant applications, particularly those involving human-computer interaction and audio search engines. As it is now understood that affective computing may have an important industry potential, automatic recognition of emotions in speech has become a more attractive area of research. For example, the automatic recognition and classification of emotions and affect in speech, based on prosodic/acoustic features, could open up exciting new possibilities for content-based information retrieval, e.g. from radio play databases.

The success of content-based information retrieval methods for audio databases depends crucially on the speech analysis algorithms used; a desideratum would be methods capable of analyzing large amounts of speech data fully automatically. The methods for extracting prosodic data automatically, especially fundamental frequency (F0), are still lacking although many algorithms have been proposed in the literature (Hess 1983).

Our general aim is to develop content based information retrieval methods for spoken Finnish, utilising the MediaTeam emotional speech corpus, the first large Finnish emotional speech database. In addition to acoustic measurements, we use subjective listening tests in order to determine how well basic emotions can be differentiated automatically vs. perceptually. In this paper, the focus is on the technical aspects and reliability of the speech analysis algorithm, F0Tool, which is used in the acoustic analysis of speech data and automatic emotion classification experiments.

F0Tool is a speech analysis software implemented in the MATLAB language; F0Tool is a cepstrum based voiced/unvoiced segmentation and time domain F0 extraction algorithm using waveform-matching. The algorithm features the following processing stages: voiced/unvoiced segmentation, F0-contour estimation and prosodic feature calculation. The algorithm is capable of analyzing large speech data sets in a single batch run or single file analysis with a GUI for visual inspection and verification purposes. The only input required by F0Tool is an audio waveform file. No phonemic annotations etc. are used. In the following three sections, the principles of F0Tool are described in some detail.

2. Voiced/unvoiced segmentation

A digital audio signal is first partitioned into 60ms overlapping segments in 6ms steps. Through the guidelines described below, a ceptrum is computed for each segment, and voiced/unvoiced (V/UV) classification is performed by combining information from consecutive segments. Cepstrum peaks are estimated in two stages: first rough estimates are computed, and then more accurate values are achieved.

For computing rough estimates, a cepstrum is computed for each segment. An amplitude correction by linear weighing is performed on the cepstrum in the range of 1/700 – 1/40 quefrencies in order to compensate for F0 variation within a segment (Noll 1967). This operation enables F0-independent global thresholding for peak detection, to be described next. It also enables better F0 estimation at the end points of voiced segments. To emphasize the F0 peaks of a noisy cepstrum and thus make peak detection more reliable, a running average liftering over the cepstrum is performed. Medians of the cepstrum peak amplitudes and segment root-mean square (RMS) energies over the speech recording are calculated next.

These medians are used in the second stage as thresholds to find the cepstral peak locations of voiced segments. If multiple peaks are present within a segment the one lowest in quefrency is selected. The peak detection operation is embedded in an F0 tracking routine that uses a 2ms tolerance window for locating the next expected pulse peak in the signal segment. This function is designed to enhance the processing of trailing voiced segments (Ahmadi & Spanias 1999).

A common problem that makes F0 estimation difficult is the frequency doubling caused by higher formants of speech. This problem was solved by applying a nonlinear function developed for this purpose to the signal amplitude prior to cepstrum calculations. By flattening the spectrum it reduces the predominance of higher formants (Hess 1983). This algorithm also improves glottal pulse peak determination of creaky voiced segments that are common in Finnish speech by stabilizing the amplitude of consecutive cycles.

Finally, a segment is classified as voiced if the RMS energy and the cepstral peak amplitude for the current segment and the one immediately preceding it are higher than the corresponding median-based thresholds. Consecutive voiced segments and segments with only one unvoiced segment between them are then joined to form the final V/UV segmentation data (Ahmadi & Spanias 1999).

3. F0-contour estimation

A waveform-matching algorithm is used to estimate the F0 pitch contours for each voiced segment (Titze & Haixiang 1993). Accurate F0 estimation is required in order to estimate features like jitter and shimmer.

First, a finite impulse response (FIR) band-pass filter is adapted to the F0 distribution obtained from the rough pitch information during V/UV segmentation from cycle period information of cepstrum peak locations. A zero-crossing calculation is then used to construct rough cycle boundaries for the waveform-matching algorithm.

Every consecutive pair of roughly marked cycles of 1kHz FIR low-pass pre-filtered data is then screened for maximum or minimum peak match using least squared error method with quadratic peak interpolation. The raw cycle peak frequency contour is then screened for simple errors and fitted with a cubic smooth contour for prosodic parameter calculations (Hess 1983).

4. Prosodic feature computation

From the V/UV segmentation and F0 contour data a total of 43 prosodic features can be calculated fully automatically. These are listed in Table 1.

Table 1. List of prosodic features

Mean F0 frequency (Hz)
Median F0 frequency (Hz)
99% value of F0 frequency (Hz)
1% value of F0 frequency (Hz)
1% - 99% F0 frequency range (Hz)
95% value of F0 frequency (Hz)
5% value of F0 frequency (Hz)
5% - 95% F0 frequency range (Hz)
Average F0 rise during continuous voiced segment (Hz)
Average F0 fall during continuous voiced segment (Hz)
Average F0 rise steepness (Hz/cycle)
Average F0 fall steepness (Hz/cycle)
Max rise during continuous voiced segment (Hz)
Max fall during continuous voiced segment (Hz)
Max steepness of F0 rise (Hz/cycle)
Max steepness of F0 fall (Hz/cycle)
Normalised segment frequency distribution width variation
F0 variance
Trend corrected mean proportional random F0 perturbation
Mean RMS intensity
Median RMS intensity
Max RMS intensity
Min RMS intensity
Intensity range
95% value of intensity
5% value of intensity
5%->95% intensity range

(Table 1 continued)

Normalised segment intensity distribution width variation
Intensity variance
Mean proportional random intensity perturbation
Average length of voiced segments
Average length of unvoiced segments shorter than 300ms
Average length of silence segments shorter than 250ms
Average length of unvoiced segments longer than 300ms
Average length of silence segments longer than 250ms
Max length of voiced segments
Max length of unvoiced segments
Max length of silence segments
Percentage of unvoiced segments shorter than 50ms
Percentage of 50-250ms unvoiced segments
Percentage of 250-700ms unvoiced segments
Ratio of speech to long unvoiced segments
Ratio of voiced to unvoiced segments
Ratio of silence to speech (speech = voiced + unvoiced<300ms)
Proportion of Low Frequency Energy under 500Hz
Proportion of Low Frequency Energy under 1000Hz

The features include basic F0-related features and derivatives (mean, median, maximum, range, fractiles, variance) and local and global derivatives of basic intensity features (mean, median, maximum, minimum, range, fractiles, variance), ratios for voiced, silent and unvoiced segments, spectral features (proportion of low-frequency energy for voiced segments) as well as some other features (trend corrected shimmer and jitter).

It should be stressed that F0Tool calculates the parameters fully automatically; the parameters do not represent any phonologically relevant features. Thus the algorithm does not operate on such features as sentence stress or intonation, although, of course, the prosodic primitives, forming the acoustic substratum of prosodic contrasts, are in some indirect ways related to phonologically relevant prosodic categories.

5. Performance of F0Tool

The performance of F0Tool was tested with a data set of 40 randomly selected sentences from radio conversations involving 4 male Finnish fighter pilots. Recordings were made using Sony TCD-D8 DAT recorders during combat training missions. The test data, including severe noise, variable speech rates and a large F0 range, was more demanding than the research data. The data was first pre-processed with a simple 55 Hz FIR high-pass filter to remove any DC offset and slow baseline drift. Audio files were then read to the F0Tool software for automatic computation of F0 contours. Software settings include various user-definable parameters like classification thresholds, analysis windowing, F0 search ranges and filters. A default setting was experimentally defined to work reliably with a set of speech corpuses and was subsequently used throughout the analysis unchanged. For example, the setting is the same for both genders. Change(s) in the settings to match a database better would cause a slight increase in absolute performance but this was not required to reach a good overall performance level.

In order to assess the accuracy of F0Tool, a reference F0 contour was defined according to the guidelines discussed in Hess (1983). An interactive software tool with a graphical user interface was prepared to this end. The user selects interactively small windows over a speech signal around the peaks corresponding to the maximum amplitude of each glottis cycle. The software then utilizes a waveform matching algorithm to locate the exact peak positions in consecutive cycles. From the peak interval values, the local F0 values are computed. This process is repeated for all voiced sections in the samples. The process includes careful inspection of each decision made by the software in order to yield as accurate F0 values as possible. A total of 6501 reference measurements of F0 were finally accepted for further analysis.

The F0 contours computed by F0Tool and the reference contours were aligned by their common time axis. Differences between the cycle lengths of the reference cycles and the corresponding estimated cycles were then computed. Mean relative error between the reference and estimated F0 values, and its standard deviation were 1.0 % and 3.4 %, respectively. Superimposition of the curves revealed that practically all of the largest errors were situated at the beginnings and ends of voiced segments.

The criteria given in Bagsahw et al. (1993) were also tested. In this scheme a deviation of less than 20 % from the reference F0 value is classified as a fine error and other deviations as gross errors. The scheme resulted in a mean relative error and its standard deviation of 0.8 % and 2.2 %, respectively, for fine errors. Of all measurements 0.22 % was classified as gross high and 0.34 % as gross low errors, for a total of 0.56 % gross error rate.

6. Discussion

The results of the reliability tests indicate that F0Tool is a reliable software for the automatic prosodic analysis of large quanta of speech data. For a highly noisy data the mean relative error of 1.0% the and low gross error of 0.56% are more than acceptable while calculating the basic prosodic parameters in a large time window. For high-precision perturbation measurements the indicated mean relative error of 0.8% for fine errors is not accurate enough to reliably measure the jitter parameter, as this would correspond to an accuracy of at least 80% for 1% perturbation. For clean speech, however, it is expected that the waveform matching algorithm will reach a mean relative error of at least 0.1% where the jitter measurements become accurate for 1% perturbation at 10% confidence (Titze & Haixiang 1993).

References

Ahmadi, S.; Spanias, A.S. 1999. Cepstrum-based pitch detection using a new statistical V/UV classification algorithm. In: *IEEE Transaction on Speech and Audio Processing* 3. 333–338.

Bagshaw, P.C.; Hiller, S.M.; Jack, M.A. 1993. Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching. In: *Proceedings of the 3rd European Conference on Speech Communication and Technology*. Berlin. 1003–1006.

Hess, W. 1991. Pitch determination of speech signals: algorithms and devices. Berlin: Springer-Verlag.

Noll, A.M. 1967. Cepstrum peak determination. In: *Journal of the Acoustical Society of America* 41. 293–309.

Titze, I.R.; Haixiang, L. 1993. Comparison of F0 extraction methods for high-precision voice perturbation measurements. In: *Journal of Speech and Hearing Research* 36. 1120–1133.

EERO VÄYRYNEN is working as a researcher at MediaTeam, University of Oulu, dealing with language and audio technologies. His research interests concern automatic prosody analysis and speech signal processing in general. He is currently working on his Master's Thesis to be finished in spring 2005 focusing on emotional content recognition with prosodic features. E-mail: eero.vayrynen@ee.oulu.fi.

HEIKKI KERÄNEN is currently working at the Department of English, University of Oulu. His research interests include phonetics and language technology, most recently tools for the transcription and annotation of speech. He has just finished his Master's Thesis on the characteristics of military pilot communication. E-mail: hkera@paju.oulu.fi.

TAPIO SEPPÄNEN is a professor of biomedical engineering of the University of Oulu. His research interests include digital signal processing and pattern recognition with applications to speech and physiologic signals, and multimedia signal processing. E-mail: tapio.seppanen@ee.oulu.fi.

JUHANI TOIVANEN studied English, phonetics and linguistics and graduated from the University of Oulu in 1992. He obtained his PhD in English philology in 1999. Juhani Toivanen has been a lecturer in general and applied linguistics and phonetics; currently, he is an Academy researcher (Academy of Finland). His research interests include prosodic analysis of first and second language, automatic recognition of emotion from speech and discourse/conversation analysis. E-mail: mailto:juhani.toivanen@ee.oulu.fi.

ESTONIAN SPEECH-DRIVEN PC-INTERFACE FOR DISABLED PERSONS

Leo Võhandu, Tanel Alumäe, Bert Viikmäe, Kairit Sirts
Rehabilitation Technologies Scientific Laboratory, Tallinn University of Technology, Estonia

Abstract

There exists already a pretty useful recognizer for Estonian numerals (Alumäe et al. 2004). There is also a long way to the next stage – full Estonian natural language speech recognizer. At the Rehabilitation Technologies laboratory of TUT we suggest to use this number recognizer to build different limited vocabulary driven but flexible and general enough input and control systems for Disabled Persons. We work in three directions. Disabled persons have often not the best possible pronounciation, so one has to build a system which can work with a lot of background and inner noise. Another direction is to build an agile table-driven 3-layer control system according to Kolmogorov-Vilenkin superposition theorem. We try also to implement a general table-driven interface description system using the finite state machine method and its very effective implementation in J-language. The solutions have to be general enough to handle easily very different disabilities needs and allow such systems not to be so expensive. In our talk we will give an overview and demonstrate our solution prototypes.

Our work has been supported by ESF grants 4844 and 5918.

Keywords: speech recognition, speech-driven interface, finite state machine

1. Introduction

There exists already a pretty useful recognizer for Estonian numerals (Alumäe et al. 2004). There is a long way to the next stage – full Estonian natural language speech recognizer. At the Rehabilitation Technologies laboratory of TUT we suggest to use this number recognizer to build different limited vocabulary driven but flexible and general enough input and control systems for Disabled Persons.

This software prototype is an agile table-driven 3-layer control system according to the Kolmogorov-Vilenkin superposition theorem. Its purpose is to help disabled persons perform basic actions with a PC. It is written entirely in Java programming language and it uses a limited-vocabulary Estonian Speech Recognition engine for recognizing numerals, which is also written in Java (Alumäe et al. 2004). The application has 3 layers of tables with controllable functionality. Behind a table there is a finite state machine, so every cell of a table is actually a state. Every state has its own functionality.

2. Input types

The prototype can be controlled by hand or by voice. Controlling it by hand means using a mouse or a floating cursor. The floating cursor is implemented as a cursor moving across the screen line-by-line and it stops when the user presses a certain button – this kind of control type is useful for persons, who can not move their hands very well. Voice recognition enables to select table cells entirely without hands by saying numbers (e.g. "23" means cell from the second row and the third column of the table).

3. Functionality

There is a main table which can be extended up to the third layer. The main table has the most generic functions and by opening a certain cell it gets more detailed. By default some functionality is already implemented in the tables, it includes basic Windows applications, Bliss language symbols and an on-screen keyboard.

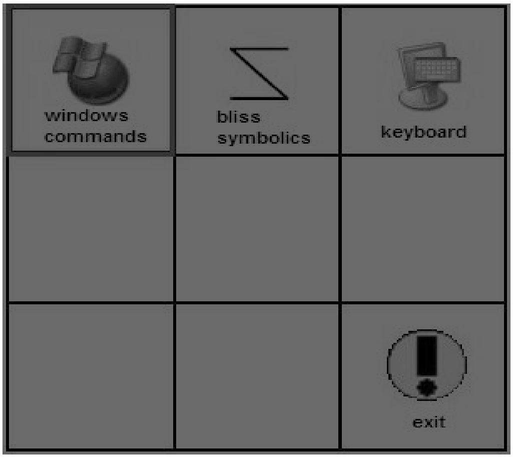


Figure1. Main table

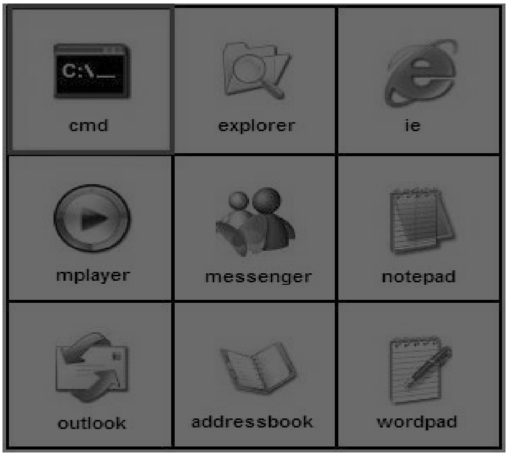


Figure 2. Windows table

4. Changing the content

Changing the content of every cell is a simple process. Each table can contain from one to 100 cells, and each cell can be edited. To do so, you just right-click on a cell and a pop-up menu appears with actions such as *create, change and delete*, by choosing one a window opens and all the editing of the cell can be done from there. A cell can contain an image, a text and functionality.

It is also possible for some of the cells to be categories and by opening them a new table appears with functionality, which belongs to this category, for example when opening a Bliss symbol called “Cooking”, a table opens with symbols related to cooking. Bliss symbols enable building sentences and speak phrases by selecting symbols.

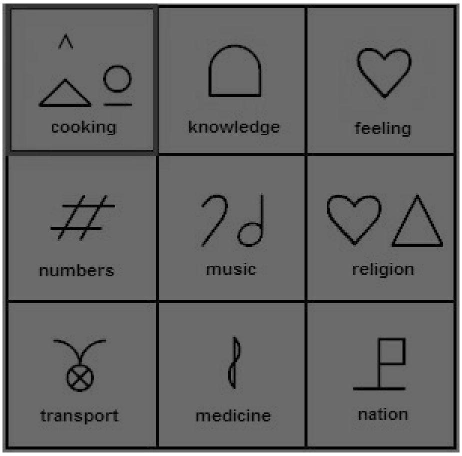


Figure 3. Bliss symbols

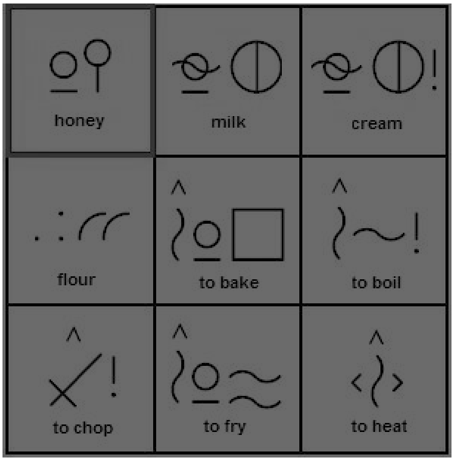


Figure 4. Bliss „Cooking“ symbols

5. On-screen keyboard

The on-screen keyboard allows access to most windows features. The most common use is typing into other programs: word processing, writing emails and surfing the internet. In addition to that it is also possible to gain control over all Windows functions - as well as emulating the keyboard, scanning through menus, including the start menu, moving and clicking the mouse pointer, launching programs and even shutting down or restarting the computer.

Custom tables are included for Word, Outlook and Internet Explorer, and new ones can be created for any other program.

6. Configuration

In the application's configuration window it is possible to choose the control type (mouse, floating-cursor, voice or all of these) and the size of the window and also the size of the table cells. The appearance of the prototype can also be changed. Colors, fonts, picture sizes, cell spacing and other options allow us to choose the programs appearance.

7. Control system implementation with finite state machines in J

7.1. Inputs

In order to use a finite state machine (fsm) implementation for control systems, we need to organize the system inputs to the machine in a proper manner. Normally, in J, the allowed fsm input alphabet can consist of desired set of consecutive integers starting from zero. So we can organize our control system interface in the way that every control information piece is presented with a number, which must be spoken by the user and recognized by the interface. We can organize the control system into different layers, where each layer augments the system possibilities and each layer uses the same set of numbers for the system control.

If we present the control interface visually for example in a 3x3 grid having 9 pieces of control information and augment it in 2 grid cells with 2 2x3 grids, then we need to number the main grid cells from 1 to 9 and augmented grids cells both from 1 to

6. It might be reasonable to leave the input alphabet symbol 0 for some inner procedures of the program.

7.2. System control as fsm control table implemented in J

Once the input set is known we must describe the fsm control table (state-transition table). Therefore we must choose the states, which in J also must be described as a set of consecutive integers starting from 0. In such a case, where fsm is used to implement the system control, it is clear, that the set of activities described by the interface and to be activated, is finite and not very big (otherwise probably usability would suffer) and therefore it is possible to use a separate state for each possible activity. By making the control table, we gather the control information from all grids into one table and thus we must use different states for each different activity in arbitrary grid and cannot use the state number overlapping as we did with input numbers. The final step for creating the control table is to describe the transitions from one state to another with each input symbol.

In our example with 3x3 grid and 2 2x3 augmented grids, assuming that each grid cell represents a different activity, we need for our control table 21 states. So the table altogether will be a 10x21 table (as we use input symbol 0 for some inner purpose).

7.3. Outputs

As each state in the control table represents an activity, we need to gather all the states the machine passed through during its processing into a list and this list is the output for the machine. For user outputs, which are the desired activities, we must take this list, translate it into a sequence of orders the user gave to the system and execute the orders in the sequence specified by the machine output list. For doing this, probably yet another program is needed, which translates the numerical states into activities, defines and executes them with a single case sentence.

7.4. What else must be considered?

There must be found a mechanism, how to start and stop the system. In J, the fsm is described with its initial configuration and if executed, it takes the initial configuration, existing input symbol sequence and processes it based on the configuration. For a control system implementation we need that the machine will stay in the (so to say) working mode after it has processed the existing input sequence as there might be further commands from the user a little bit later. We need to find the way, how not to reset the machine into the initial configuration after one input sequence has been processed. Probably the solution in this case would be to give the machine a new set of initial configuration after each input sequence processing based on the state, where the machine currently stopped its work.

There is also another issue, which we are actually not considering now. There are great many different fsm constructs available. One of the most common divisions of fsm-s is its division into Mealy and Moore machine. J gives the good possibilities for Moore machine implementation, but unfortunately, it is not a good solution in all cases. As long as we are doing some control processing or processing some data, which ends up with a very small set of different data, the Moore machine seems to be just fine. But as our system is expected to process data, with results in a rather large data set, the Mealy machine would be a better choice. In this case, the states represent the states of data rather than the data itself.

References

Alumäe, Tanel; Võhandu, Leo 2004. Limited-Vocabulary Estonian continous speech recognition system using Hidden Markov Models. In: *Informatica* 15, No.3. 303–314.

LEO VÕHANDU, prof. emer., head of the Rehabilitation Technologies Scientific Laboratory, Tallinn University of Technology. He recieved his Ph.D in Applied Mathematics at the Tartu University in 1955. His main interests lie in Data Analysis. He has written a couple of monographs and hundreds of articles. Under his guidance over 25 Ph.D dissertations have been defended. Now his main interest lies in the field of using IT for Disabled Persons Rehabilitation. E-mail: leov@staff.ttu.ee

TANEL ALUMÄE is a doctoral student of the first author. (About personal data, see his article at this Conference.)

BERT VIIKMÄE is a master student of the first author.

KAIRIT SIRTS is a master student of the first author.

THE MORPHOLOGICALLY ANNOTATED LITHUANIAN CORPUS

Vytautas Zinkevičius, Vidas Daudaravičius, Erika Rimkutė
Vytautas Magnus University (Kaunas, Lithuania)

Abstract

The paper deals with the preliminary findings from the morphologically annotated corpus of Lithuanian language (1 million running words). It was compiled and processed at the Center of Computational Linguistics, Vytautas Magnus University. Each annotation for an inflected word form of the corpus contains a lemma and a set of morphological features. The paper presents the strategy for automatic and manual annotation. Automatic annotation was carried out with the help of analyser-lemmatiser. Disambiguation of the homoforms was performed manually. Tag sets and the most prominent features of Lithuanian morphology are discussed in detail. The annotated corpus allowed us to measure the usage of parts of speech and their morphological features in contemporary Lithuanian language. The annotated corpus is of great importance for future development of parsing tools, treebanks and other NLP tools and resources for Lithuanian language.

Keywords: corpus compilation, morphological annotation, Lithuanian language, tag sets, morphological disambiguation

1. Introduction

The paper describes processing of the first Lithuanian annotated corpus (LAC) at the Center of Computational Linguistics (CCL), Vytautas Magnus University. LAC is important since it enables linguistic, statistical and computer science research. Moreover, it gives the ground for future development of language technologies for Lithuanian, e.g. automatic morphological disambiguators, parsing tools, treebanks. Texts for compiling LAC were selected from the Corpus of the Contemporary Lithuanian Language at CCL (more about see Marcinkevičienė et al. 2004), which comprises more than 100 million running words.

LAC is a set of XML files, containing 1 million running words annotated morphologically. Each annotation for a word form contains its normalized form (lemma), and a full set of morphological properties for the inflected word form. Non-word textual units, such as punctuation marks, spaces, paragraphs, numbers, are represented in LAC by special marks.

The annotation of the LAC started at 2001. In the beginning the collection of the texts was compiled. The collection had to present a wide range of textual genres and registers in order to have as much of the variety of morphological information as possible, e.g. scientific texts, fiction, parliament debates and administrative texts, to mention a few.

Table 1. The set of POS tags and their abbreviations

Grammatical Category	Equivalent in English	Tag
Abbreviation	dr.	Abbr
Acronym	NATO	Acronym
Adjective	good	Adj
Adverb	perfectly	Adv
Onomatopoetic interjection	cock-a-doodle-do	Onom
Conjunction	and	Conn
Half participle	when speaking	Half_part
Infinitive	to be	Inf
Second Infinitive	at a run	Inf2
Interjection	yahoo	Interjection
Noun	a book	N
Number	one	Numb
Roman Number	I	Numb2
Proper Noun	London	PN
Proper Noun2	Don	PN2
Participle	walking	Part
Gerund	on the walk home	Gerund
Preposition	on	Prep
Pronoun	he	Pron
Verb	do	V
Idiom AA	rest eternal	idAA
Connective idiom	et cetera	idConn
P.S.	P.S.	idPS
Prepositional idiom	inter alia	idPrep
Pronominal idiom	nevertheless	idPron
Particle	also	Particle

Table 2. The set of the morphological features and their abbreviations

Property	Value	Tag
Reflexiveness	reflexive	Ref
	non-reflexive	NonRef
Positiveness	positive	Pos
	negative	Neg
Voice	active voice	ActVoice
	passive voice	PassVoice
	participle of necessity	PartOfNecess
Mood	indicative mood	IndicatM
	imperative mood	ImperatM
	subjunctive mood	SubjunctM
Tense	present tense	PresT
	simple past tense	SimplePastT
	past frequentative tense	PastFreqT
	past tense	PastT
	future tense	FutT
Numeral type	cardinal numeral	CardinalNumb
	plural numeral	PlurNumb
	collective numeral	CollectNumb
	ordinal numeral	OrdinalNumb
Degree	positive degree	PosDeg
	comparative degree	CompDeg
	attenuated degree	AttenDeg
	superlative degree	SupDeg
Definiteness	definitive	Def
	undefinitive	Undef
Gender	common gender	Comm
	masculine gender	Masc
	feminine gender	Fem
	neuter gender	Neut
Number	singular	Sg
	plural	Pl
	dual	Dual
Case	nominative	Nom
	genitive	Gen
	dative	Dat
	accusative	Acc
	instrumental	Inst
	locative	Loc
	vocative	Voc
	illative	Il
Person	first person	I
	second person	II
	third person	III

LAC is the first Lithuanian morphologically annotated corpus that includes full texts of various genres and registers. Nevertheless a few databases containing data about lexical and inflectional frequencies could be mentioned here as being closest products for Lithuanian language (Grumadienė 2002; Mauricaitė et al. 2004). In comparison with annotated corpora for other languages LAC is of similar size as many other manually annotated corpora but different in annotation level. For instance, the *TIGER* corpus of German has 1 million words (Brants et al. 2002), the treebank of Russian has 1 million words (Boguslavsky et al. 2000), the treebank of Czech consists of 1,5 million words (Hajič 2002). These above-mentioned corpora are annotated morphologically and syntactically. Prague Dependency Treebank is also annotated tectogrammatically (Hajičová 1998). Tag set of each corpus is different in size and in features. Mostly it depends on the type of the language, i.e. inflected languages have a richer tags sets.

2. Corpus Annotation

The Lithuanian language is a highly inflected language, e.g. ending *-o* is grammatically polysemous since it denotes singular Genitive of masculine noun, pronoun or numeral, e.g. *šito vieno aukšto perstatyto pastato (of this one floor rebuilt house)*. This ending can also be the third person of present or past tense verb form, e.g *daro* (*he does/they do*), *ėjo* (*he/they went*). Tables 1 and 2 show the system of tags used for the morphological annotations in LAC.

The example from LAC shows the structure of the tags and text annotation. “word” presents the word form or token used in the text, “lemma” is the normalized form of the word form and “type” is the set of morphological features which describes the word form.

The example from LAC:

```
<word="Skulptūra" lemma="skulptūra" type="N Fem Sg Nom">
<space>
<word="buvo" lemma="būti(yra,buvo)" type="V Pos NonRef ActVoice SimplePastT Sg III">
```

Every part of speech has its specific set of tags, e.g. noun is usually described with a help of three features: gender, number and case. However some loan words lack some features, e.g. the word “taxi” in Lithuanian is not inflected and it is difficult to assign gender, number and case. In this case no features, only part of speech tag is assigned to “taxi”, i.e., <word=”taxi” lemma=”taxi” type=”N”>. Some morphological multiword units that can not be analyzed separately required specific additional tags such as pronominal idioms, prepositional idioms, connection idioms. Besides one additional case tag for the obsolete illative case missing in contemporary Lithuanian language was included.

Non-lexical units were marked using separate tags, e.g., spaces were marked as “<space>”, all punctuation marks were put in the tag <sep=”…”> (e.g. <sep=”!”>), numbers were put in the tag <number=”…”> (e.g. <number=”86”>), foreign insertions were marked between “<foreign lang=”…”>” (e.g. <foreign lang=”en”>) and “</foreign>”.

The process of LAC annotation is presented in Figure 1. Possible lemmas and the morphological features were automatically assigned to all word forms using the morphological analyzer-lemmatizer Lemuoklis (Zinkevičius 2000). Since the tool processes only isolated words, it produces all theoretically possible grammatical interpretations for each inflected word form of text, including all possible lemmas. Not all word forms were recognized therefore manual assignment of lemmas and morphological features was necessary. Lemmatization was followed by manual disambiguation since almost half of the word forms were ambiguous in the morphological sense.

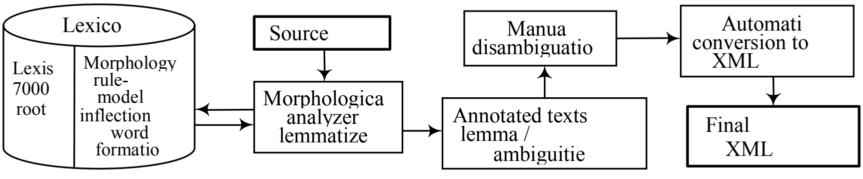

Lexico
Lexis 7000 root
Morphology rule- model inflection word formatio
Source
Morphologica analyzer lemmatize
Annotated texts lemma / ambiguitie
Manua disambiguatio
Automati conversion to XML
Final XML

Figure 1. The process of LAC annotation

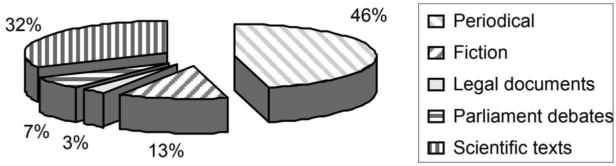


Figure 2. The textual structure of LAC

3. Morphological ambiguity and the resolution

Manual disambiguation revealed that circa 47 % of Lithuanian words or word forms are morphologically ambiguous. The morphological ambiguity of inflected languages is similar, e.g. the morphological ambiguity of Czech language is ca 46 % (Hajič 2004: 173). There are two types of morphologically ambiguous words or word forms: lemma ambiguity and word form ambiguity. An example of lemma ambiguity is *namo* – noun singular Genitive (*namas*) and adverb (*namo*). An example of word form ambiguity is *mamos* (*mother's* or *mothers*) – singular Genitive or plural Nominative.

Lemuoklis can not recognize ca 11 % of word forms of the corpus. The most typical unidentified cases are: proper nouns, foreign words, abbreviations, acronyms, abbreviated forms, etc. Only ca 40 % of all word forms are morphologically unambiguous, e.g. *aš (me/I)* always has one lemma and one morphological tag (Rimkutė 2003: 60–78).

4. The structure of the LAC and the grammatical specificity of the Lithuanian language

The LAC contains ca 1 013 000 running words, ca 145 000 types (word forms), ca 49 000 lemmas. The structure of the LAC is presented in Figure 2. The average number of the word form/lemma ratio is 3.6. This means that one lemma has a bit more then three word forms in average. This ratio for Lithuanian language is much higher in comparison with the same for Polish that is equal 2.01 (the ratio was calculated on IPI PAN corpus (Przepiórkowsky 2004)). This means that Lithuanian word forms are much more inflected than Polish, despite the fact that not all parts of speech are inflected (see Figure 3). Only six parts of speech out of eleven are inflected: noun, verb, pronoun, adjective, participle and number. The most inflected part of speech is pronoun which lemma has up to seven word forms in average. Since pronouns cover a considerable part of the usage of contemporary Lithuanian (see Figure 4 for details) this means that pronouns are one of the most difficult parts of speech for grammatical analysis.

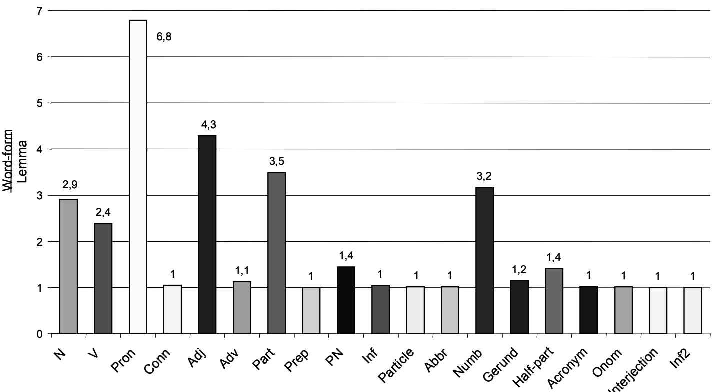


Figure 3. Word form/lemma ratio for part of speech

The largest amount of usage of Lithuanian language is dominated by nouns i.e. more than 36 %. Finite and non-finite

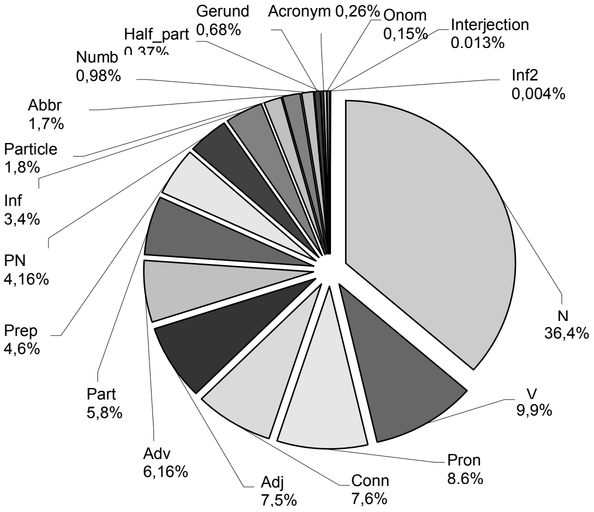


Figure 4. Distribution of parts of speech in LAC

verbs (infinitive, participle, half-participle, gerund, second infinitive) make up 20 % of the usage of Lithuanian language.

5. Concluding remarks

Annotation of the first Lithuanian language corpus provided with some valuable data concerning the grammatical specificity of the Lithuanian language. It turned out that inflection in real usage is not so prominent as in the grammatical system since highly inflected parts of speech such as verbs and nouns have less than 3 word-forms in average. Pronoun demonstrated surprisingly big number of word forms actually used in contemporary Lithuanian language. Overall, the tendencies for usage of different parts of speech coincide with the data obtained by other researchers as well as native speaker's intuitions, i.e. noun has the biggest coverage but verb is also very important. Thus Lithuanian language preserves its verbal nature.

The next step for processing of LAC will be its syntactic annotation with a help of Dependency Grammar formalism. LAC will be also used as a machine learning basis for the annotation of 100 million word corpus. Finally, LAC will be available on the web for research purposes.

Acknowledgement

We are grateful to the State Commission of the Lithuanian Language for supporting the compilation of the LAC in 2001–2003.

References

Boguslavsky, I.; Grigorieva, S.; Grigoriev, N.; Kreidlin, L.; Frid, N. 2000. Dependency treebank for Russian: concept, tools, types of information. In: *Proceedings of 18th International Conference on Computational Linguistics (COLING-2000).* Saarbrüken, Germany. Vol 2, 987–991.

Brants, Sabine; Dipper, Stefanie; Hansen, Silvia; Lezius, Wolfgang; Smith, George 2002. The TIGER treebank. In: *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Sozopol, Bulgaria. Retrieved February 15, 2005, from http://www.coli.uni-sb.de/~sabine/tigertreebank.pdf.

Grumadienė, Laima 2002. Dabartinės rašomosios lietuvių kalbos dažninis žodynas ir jo bazė [The frequency dictionary of contemporary written Lithuanian and its data base]. In: Holvoet, A. (ed.) *Acta Linguistica Lithuanica* 46. Vilnius: Lietuvių kalbos institutas (Institute of the Lithuanian Language). 19–37.

Hajič, Jan 2002. Tectogrammatical representation: towards a minimal transfer in machine translation. In: *Proceedings of the Sixth International Workshop on*

Tree Adjoining Grammar and Related Frameworks (TAG+6). Venezia, Italy: Universita di Venezia. 216–226.

Hajič, Jan 2004. Disambiguation of rich inflection. Computational morphology of Czech. Prague: Karolinum Charles University Press.

Hajičová E. 1998. Prague Dependency Treebank: from analytic to tectogrammatical annotations. In: Sojka, Petr; Matoušek, Václav; Pala, Karel; Kopeček, Ivan (eds.) *Text, Speech, Dialogue*. Brno: Masarykova univerzita. 45–50.

Marcinkevičienė, R.; Bielinskienė, A.; Daudaravičius, V.; Rimkutė, E. 2004. Corpora for Lithuanian language technologies. In: *The First Baltic Conference. Human Language Technologies. The Baltic Perspective*. Riga, Latvia, April 21–22, 2004. 21–24.

Mauricaitė, Vera; Norkaitienė, Milda; Pakerys, Antanas; Petrokienė Ritutė 2004 (eds.). Bendriniai XX a. spaudos žodžiai. Elektroninis dažninis žodynas [Common press words of the 20th century. Electronic frequency dictionary]. Vilnius: MELI [Science & Encyclopaedia Publishing Institute].

Przepiórkowski, Adam 2004. The IPI PAN corpus: preliminary version. Warszawa: Instytut podstav infromatyki PAN. Retrieved February 17, 2005, from http://dach.ipipan.waw.pl/~adamp/Papers/2004-corpus/book_en.pdf.

Rimkutė, Erika 2003. Morfologinio daugiareikšmiškumo tipologija [The typology of morphological ambiguity]. In: Merkys, V.; Ambrazas, V.; Sauka, L. (eds.) *Lituanistica* 4 (56). 60–78.

Zinkevičius, Vytautas 2000. Lemuoklis - morfologinei analizei [Morphological analysis with Lemuoklis]. In: Gudaitis, L. (ed.) *Darbai ir Dienos* 24. 246–273.

VYTAUTAS ZINKEVIČIUS is engineer programmer at the Center of Computational Linguistics, Vytautas Magnus University. He has graduated from Vilnius University at 1981. His research interests concern natural language resources, theory and tools for language analysis and generation, computational morphology for highly inflected languages, computer readable lexicons. His doctoral study focuses on digital modeling of the Lithuanian morphology. He is a developer of computational models for morphology and lexis that were implemented in several Lithuanian spell-checkers, and in compiling frequency dictionaries for contemporary Lithuanian. He also participates in creating the digital version of Dictionary of Lithuanian in 20 vol., an investment project at the Institute of Lithuanian Language. E-mail: vytasz@lki.lt

VIDAS DAUDARAVIČIUS is senior engineer-programmer and early-stage researcher at the Centre of Computational Linguistics Vytautas Magnus University. He received his M.A. (Applied Informatics) at VMU Faculty of Computer Science. His fields of research: computational syntax and parsing, computational linguistics, information retrieval and extraction, machine translation.

ERIKA RIMKUTĖ is a junior researcher of the Centre of Computational Linguistics at Vytautas Magnus University. She received her M. A. (Lithuanian language) at VMU. Her research interests include corpus linguistics, computational linguistics, automatic morphological analysis and synthesis, morphological ambiguity and disambiguation and automatic syntactic analysis. Her doctoral study focuses on morphological ambiguity and disambiguation in the Lithuanian language. E-mail: e.rimkute@hmf.vdu.lt.

A SPEECH DATABASE FOR UNIT SELECTION SLOVENIAN TEXT-TO-SPEECH SYNTHESIS

Jerneja Žganec Gros[1], Aleš Mihelič[1], Nikola Pavešić[2], Mario Žganec[1], Varja Cvetko-Orešnik[3], Primož Jakopin[3]

[1]Alpineon RTD, Iga Grudna 15, SI-1000 Ljubljana, Slovenia
[2]University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia
[3]Fran Ramovš Institute of the Slovenian language, Novi Trg 2, Ljubljana, Slovenia

Abstract

The paper focuses on the design and collection of a speech corpus of elemental speech units for AlpSynth, a corpus-driven Slovenian TTS system. We describe the design procedures for a new speech corpus: purpose definition, content selection, definition of recording conditions and requirements, corpus segmentation and annotation.

First we describe and comment the results of a frequency analysis of Slovenian allophone strings (diphones, triphones, quadphones) performed on a large Slovenian input text that has been converted to allophones using grapheme-to-allophone conversion rules for the Slovenian language.

Further we present a method we designed for selection of a compact and efficient set of Slovenian sentences out of a large text corpus so as to minimize the final representative speech corpus. The selected sentences cover all the desired most frequent Slovenian quadphones, triphones and subsequently diphones.

We describe the recording sessions and recording conditions. We continue describing the corpus annotation process – semiautomatic – and comment the annotation tools used. Finally, we describe the archive structure of the spoken corpus and present the information on its structure, content and size.

Keywords: speech corpus, text-to-speech synthesis, Slovenian language, unit-selection TTS, allophone frequency analysis

1. Introduction

The ability of telephone speech recognition and text-to-speech synthesis to support effective applications has been established (Meisel 2002). The large number of successfully deployed speech technology applications has proven the technology works, cost savings or increased revenues have been achieved and speech recognition has introduced substantial improvements in the user interface over the touch-tone pad.

Telephone service providers accepted speech recognition and text-to-speech synthesis (TTS) as being of long-term strategic importance, W. Meisel continues in his resume of the major speech technology events. Major telecoms, e.g. Sprint PCS, AT&T, BellSouth and many others, launched and expanded speech driven services, and indicated plans for continuing to do so. After the period of huge investments into home internet phone devices the recent successful speech recognition implementations drive the industry back to black-phone strategies using speech technologies to introduce new value-added services, using simple traditional phones.

A vital part of speech technology applications in modern voice application platforms is a text-to-speech engine. Text-to-speech synthesis enables automatic conversion into spoken form of any available textual information.

The initial attempts towards Slovenian TTS were mainly based on concatenation of diphones, and they resulted in a few demonstration systems (Šef 2001), (Gros 2001), (Vesnicer et al. 2001), (Vesnicer 2003) and some first carrier-grade voice applications (Gros et al. 2002). In (Rojc et al. 2000) the authors describe the formation of only a text corpus for Slovenian corpus based TTS. The AlpSynth TTS system follows similar principles as the S5 TTS system (Gros 2001). However, it is based on concatenation of basic speech units, derived from a large speech corpus, instead of diphones only. The input text is transformed into its spoken equivalent by a series of modules.

A grapheme-to-allophone module produces strings of phonetic symbols based on information in the written text. The problems it addresses are thus typically language-dependent. A prosodic generator assigns pitch and duration values to individual phones. Pitch modeling is based primarily on predicting the proper Slovenian tonemic accent. Phone duration is predicted by a two level approach, taking into account how acceleration or slowing down affect the duration of individual phones. Final speech synthesis is based on corpus segment concatenation using TD-PSOLA (Moulines 1990).

2. Unit-selection concatenation-based TTS

Given a sequence of phonetic symbols and prosody markers derived by the prosody prediction modules, the final step within the AlpSynth TTS system is to produce audible speech by assembling elemental speech units. This is achieved by taking into account computed pitch and duration contours, and synthesizing a speech waveform.

The TD-PSOLA technique (Moulines et al. 1990) was used for speech segment concatenation as it enables pitch and duration transformations directly on the waveform, at least for moderate ranges of prosodic modifications without considerably affecting the quality of the synthesized speech.

A corpus-based approach has been used for speech synthesis for the first time in Slovenian speech synthesis. First an extensive analysis of the frequency of Slovenian polyphone sequences was performed. Large Slovenian text corpora have been transcribed into allophone sequences and statistically processed. The texts for the spoken corpus were selected by an optimization process optimizing the number of the most frequent polyphones covered by the spoken text and a minimum amount of the text to be read by the speaker.

Given an input sequence of phonetic symbols a rather sophisticated segment selection algorithm first selects the segments to be concatenated. It takes into account a number of criteria ranging from more and less preferred allophones for concatenation, the length and phonetic contexts of polyphones, spectral discontinuities, etc.

3. Speech corpus

For corpus-based text-to-speech synthesis a speech corpus of recorded and annotated elemental speech units is required (Conkie 1999), (Conkie et al. 1991). The quality of the output synthetic speech depends crucially on the quality of the speech corpus.

The longer elemental speech units are used the better and more natural-sounding synthetic speech the TTS system can yield. However, with longer elemental speech units the corpus size increases dramatically. Therefore, a compromise between the size of the speech corpus and the quality of the resulting speech has to be taken (Beutnagel et al. 1999).

The process of designing a speech corpus for corpus-driven TTS can be divided into three phases:

– representative sentence set selection,
– recording of selected texts and
– segmentation and annotation of the recorded speech material.

3.1. Representative sentence set selection

Initially, we collected a large Slovenian corpus of reference texts covering various text styles, ranging from newspaper articles to novels.

All sentences shorter than 5 words were discarded from further analysis. The remaining reference text corpus contained 200.000 different sentences (25 Mbyte of text in ASCII format). This reference text corpus was processed by a grapheme-to-allophone converter from the AlpSynth TTS system in order to obtain an allophone transcription of the reference text corpus.

Most allophone or phoneme frequency analysis experiments for the spoken Slovenian language have been performed on orthographic text transcriptions and represented a grapheme frequency analysis for the language rather than a proper phoneme or allophone frequency analysis.

The grapheme-to-allophone conversion of the reference text corpus enabled us to perform an analysis on frequent phone sequences of allophones, diphones, triphones and quadphones. The results of the analysis gave us an idea on how frequently certain phone combinations occur in the spoken Slovenian language.

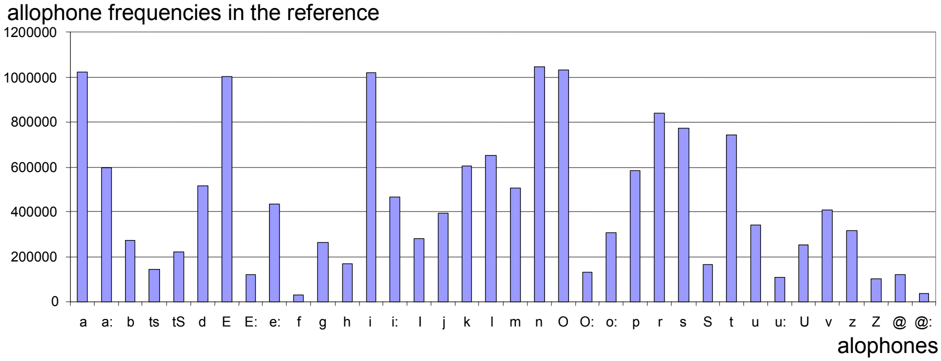


Figure 1. Allophone frequencies in the phonetic transcription of the reference text corpus. The allophones are presented in alphabetical order.

Allophone frequencies are shown in Figure 1. The allophones are presented by SAMPA symbols for Slovenian. Allophones of stressed and unstressed vowels were treated separately in the analysis. Similarly to other languages, vowels were found to be the most frequent allophones. Further, the analysis has shown that just are few triphones have very frequent occurrences, whereas some of them hardly occur at all.

Therefore it makes sense to select just the most frequent triphones to be represented in the final speech corpus. We have opted for the first 500 triphones, that represent 1% of the complete triphone set but they cover almost 50% of all triphones in the transcribed reference text corpus. In a similar way 300 most frequent quadphones were selected.

With the most frequent triphones and quadphones selected we wanted to design an optimal compact set of corpus sentences that cover all the chosen allophone sequences. A special sentence selection algorithm was designed for this purpose (Mihelič 2002). Each sentence in the reference text corpus was equipped with a cost attribute, based on the amount of the preselected frequent allophone sequences they contained. The highest cost value was attributed to a rare preselected quadphone, the lowest to a frequent preselected triphone. In order to avoid the selection of long sentences (that contain more allophone sequences than shorter sentences) the cost value was normalized with the total number of allophones within the sentence.

The sentence with the highest score was selected for the final text corpus. The preselected allophone sequences covered by this sentence were eliminated from the list. Then the cost derivation and sentence selection process was preformed for this new set of preselected allophone sequences and a new sentence was chosen for the final text corpus. The same process repeated in a loop until the all of the initial preselected allophone sequences were covered in the resulting corpus of selected sentences.

The sentence selection algorithm was capable of selecting a rather modest amount of phonetically representative sentences out of the reference text corpus. A total of 297 sentences were selected out of the initial 200.000 sentences from the reference text corpus. The phonetic transcription of the selected sentence set covered all preselected most frequent triphones and quadphones.

3.2. Recording and corpus annotation

The resulting set of sentences was chosen for the graphemic representations for the final speech corpus. These sentences were recorded along with logatoms containing all phonetically possible diphone combinations for the spoken Slovenian language (diphones containing diphtongs were treated separately).

The recorded speech material has to be segmented, annotated and pitch-marked. The SigMark speech processing tools was used for this purpose. For segmentation of elemental speech units a semi-automatic procedure was used (Dobrišek 2001). Manual corrections were performed using SigMark. The final speech corpus for corpus-based Slovenian text-to-speech synthesis contains 297 read sentences with 1814 words and 1668 logatoms.

4. Conclusion

The presented work shows the corpus-design and construction process for corpus-based text-to-synthesis. An effective method for selection of a compact set of sentences from the reference text corpus that cover most frequent allophone sequence occurrences in a given language was presented. The first attempts at developing a corpus-driven text-to-synthesis system for the Slovenian language are promising, so that further work on improving individual parts of the system is encouraged.

Acknowledgements

The authors of the paper wish to thank the Slovenian Ministry of Higher Education, Science and Technology and the Slovenian Ministry of Information Society for co-funding the project (CRP-2003 project No. V2-0896, project No. L6-5405).

References

Beutnagel, M., Mohri, M. and Riley, M. 1999. Rapid unit selection from a large speech corpus for concatenative speech synthesis. *Proceedings of the Eurospeech '99 Conference*. Budapest, Hungary.

Conkie, A. 1999. Robust unit selection system for speech synthesis. *Proceedings of the Eurospeech '99 Conference*. Budapest, Hungary.

Conkie, A., Beutnagel, M., Syrdal A., and Brown P. 2000. Preselection of candidate units in a unit selection-based Text-to-Speech synthesis system. *Proceedings of the ICSLP '00 Conference*. Beijing, China.

Dobrišek, S. 2001. Analiza in razpoznavanje glasov v govornem signalu. PhD Thesis. Faculty of Electrical Engineering. University of Ljubljana (in Slovenian).

Gros, J. 2001. Samodejno tvorjenje govora iz besedil. Linguistica et Philologica. ZRC SAZU. Ljubljana. Slovenia (in Slovenian).

Gros, J., Žganec, M., Mihelič, A., Knez, M., Merčun, A., Marinčič, D. 2002. The phonetic family of voice-enabled products. *Proceedings of the Language Technologies Conference*, Ljubljana, Slovenia. 127.

Meisel, W. 2002. Looking back at 2001 and forward to 2002. *Speech Recognition Update.* 103.

Mihelič, A. 2002. Zbirka govornih signalov za sintezo slovenskega govora. MSc Thesis. Faculty of Electrical Engineering, University of Ljubljana (in Slovenian).

Moulines, E. and Charpentier, F. 1990. Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones, *Speech Communication* 9. 453.

Rojc, M., Kačič, Z. 2000. Design of optimal Slovenian speech corpus for use in the concatenative speech synthesis system. *Proceedings of the Second international conference on language resources and evaluation*. Athens, Greece. 321.

Šef, T. 2001. Analiza besedila v postopku sinteze slovenskega govora. PhD Thesis. Faculty of Computer Science and Informatics. University of Ljubljana (in Slovenian).

Vesnicer, B., Pavešić, N. and Mihelič, F. 2001. Korpusna sinteza govora. *Proceedings of the ERK'01 Conference*. Portorož, Slovenia. Vol. B. 253 (in Slovenian).

Vesnicer, B. 2003. Umetno tvorjenje govora z uporabo prikritih Markovovih modelov. MSc Thesis. Faculty of Electrical Engineering. University of Ljubljana (in Slovenian).

Yi, J. 1998. Natural-sounding speech synthesis using variable-length units. MEE Thesis. MIT.

JERNEJA ŽGANEC GROS is head of the Alpineon research group. She received her PhD from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia. Her research interests include speech processing, pattern recognition and information communication technologies. She has authored or co-authored more than 100 papers and 2 books addressing several aspects of the above areas. She is a member of IEEE, ISCA, the Slovene Pattern Recognition Society (co-founder and first secretary) and the Slovene Language Technologies Society (co-founder and first treasurer). E-mail: jerneja.gros@alpineon.com.

ALEŠ MIHELIČ is a research member of the Alpineon research group. He received his MSc from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia. His research interests include speech processing and pattern recognition. His doctoral study focuses on text-to-speech synthesis for embedded applications. He is a member of the Slovenian Language Technologies Society. E-mail: ales.mihelic@alpineon.com.

NIKOLA PAVEŠIĆ is head of the Laboratory of Artificial Perception, Systems and Cybernetics, and Professor at the Faculty of Electrical Engineering, University of Ljubljana, Slovenia. His research interests include speech processing, computer vision, pattern recognition, biometrics and information theory. He has authored or co-authored more than 200 papers and 4 books addressing several aspects of the above areas. He is a member of IEEE, the Slovene Association of Electrical Engineers and Technicians (meritorious member), the Slovene Pattern Recognition Society (founder and first president) and the Slovene Society for Medical and Biological Engineering. He is a member of editorial boards of several technical journals. E-mail: nikolap@fe.uni-lj.si.

MARIO ŽGANEC is CEO of Alpineon RTD. He received his PhD from the Faculty of Medicine, University of Ljubljana, Slovenia. His research interests include speech processing, computer vision, pattern recognition, biomedical informatics and information communication technologies. He has authored or co-authored more than 50 papers. He is a member of the Slovene Pattern Recognition Society and the Slovene Language Technologies Society. E-mail: mario.zganec@alpineon.com.

VARJA CVETKO-OREŠNIK is head of the Fran Ramovš Institute of the Slovenian Language at ZRC SAZU and Professor at the Faculty of Arts, University of Ljubljana, Slovenia. Her research interests include Indo-European comparative linguistics, Balto-Slavic-Iranian areal linguistics, early history of Slovenian. She has authored or co-authored several publications addressing aspects of the above areas. She is a member of the Indogermanische Gesellschaft, Wiener Sprachgesellschaft. E-mail: cvetko@zrc-sazu.si.

PRIMOŽ JAKOPIN is head of the Corpus Laboratory at the Fran Ramovš Institute of Slovenian Language at ZRC SAZU and Assistant Professor at the Faculty of Arts, University of Ljubljana, Slovenia. His research interests include corpus linguistics, information theory, language modeling and lexicography. He has authored or co-authored more than 100 papers and 5 books addressing several aspects of the above areas. He is a member of the Slovenian Language Technologies Society, Association for Computational Linguistics and Association for Computing Machinery. E-mail: primoz.jakopin@guest.arnes.si.

Author Index