Max Silberztein
Editor

Linguistic Resources for Natural Language Processing

On the Necessity of Using Linguistic Methods to Develop NLP Software

Springer

Editor
Max Silberztein
Université de Franche-Comté
Paris, France

ISBN 978-3-031-43810-3 ISBN 978-3-031-43811-0 (eBook)
https://doi.org/10.1007/978-3-031-43811-0

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Part III Linguistic Resources for Low-Resource Languages

Linguistic Resources and Methods for Belarusian Natural Language Processing

Yuras Hetsevich and Mikita Suprunchuk

Abstract We present several newly developed services, methods, and algorithms for Belarusian in the field of NLP, which provide users with a set of tools for text, speech and other multimedia processing. Such services as the Speech synthesizer, Transcription generator, Word paradigm generator are grounded on a rule-based approach. The algorithms, databases and lists of rules for their development are described in detail. Each tool realizes a specific task for computational processing of textual information and speech. The proposed resources are also used to collect targeted thematic content to develop and refine natural language processing systems for Belarusian. As a consequence, we show that linguistic resources have not lost their relevance to NLP.

Keywords Automatic language processing · Belarusian · Concordance · Dictionary · Natural language processing · Phonetics · Transcription · Text-to-speech synthesizer · Word paradigm

1 Introduction

In the second half of the 2010s, methods based on the use of neural networks and deep machine learning gained popularity in the sphere of Natural Language Processing (NLP). However, these methods have some drawbacks: they require large quantities of textual materials to build their language models. Unfortunately, the amount of required textual material is not always available. In addition, a language model trained on a certain dataset may not be able to process linguistic phenomena correctly if their frequency is too low in the dataset. Machine learning requires a long running time, which developers may not have. Neural models show excellent results at processing wordforms, but results at higher linguistic levels are less impressive.

Rule-based NLP approaches are still relevant today: they achieve greater accuracy than empirical approaches when solving a lot of problems in computational

Y. Hetsevich (✉) · M. Suprunchuk
United Institute of Informatics Problems, Minsk, Belarus

M. Silberztein (ed.), *Linguistic Resources for Natural Language Processing*,
https://doi.org/10.1007/978-3-031-43811-0_4

linguistics, and the linguistic resources created for a specific NLP application can further contribute to the general collection of language knowledge.

For the last 50 years, the Speech Synthesis and Recognition Laboratory of the United Institute of Informatics Problems of the National Academy of Sciences (Minsk) has been working on developing software tools and linguistic resources to process texts and other data for the Belarusian language. In 2014, the Institute published the WEB site www.corpus.by to offer WEB services for voice, and text processing. The services are sorted into thematic groups for convenient usage in specific fields of application, as presented in Hetsevich et al. (2021). In the following, we present three services that use carefully handcrafted linguistic resources for Belarusian: the Text-to-Speech Synthesizer, the Transcription Generator, and the Word Paradigm Generator.

2 Text-to-Speech Synthesizer

Our Text-to-Speech Synthesizer system (TTS), implemented in C++, processes a written text and constructs an audio file that users can listen to, download and save. It is publicly available at: www.corpus.by/TextToSpeechSynthesizer.

Its model is based on theoretical and experimental data specific to Belarusian: the linguistic resources formalize the phonetic and prosodic structure of speech as well as articulatory and acoustic phenomena involved in speech formation. TTS uses a multi-wave approach to synthesis, *i.e.*, it compiles segments of a natural speech wave, correlated with elements of various phonetic lengths: allophones, diphones, and three-phones.

TTS uses a handcrafted phonetic-acoustic database that describes the intra- and inter-language specific to phonetic systems, as well as positional-combinatorial phenomena that generate allophonic speech, cf. (Taylor 2009). TTS performs lexical and grammatical analysis of the input text by modeling the process of speech formation, considering pronunciation and intonation features of Belarusian. The input text undergoes a sequence of processing performed by specialized processors: a text processor, a prosodic processor, a phonetic processor, and an acoustic processor; each processor is associated with a specific database that contains handcrafted rules. For more details see (Lobanov and Tsirulnik 2008).

- The text processor processes the input text in the following sequence: text cleaning, character conversion (abbreviations, acronyms, numbers, etc.), placement of accents, and POS tagging of wordforms. The resulting annotated text is then processed by the prosodic processor, which divides it into syntagmatic phrases and accent units (AU), which are further marked up into elements: pre-core, core, and post-core intonation, and then sets the values of amplitude (A), phoneme duration (D), and pitch frequency (F0) for each AU, in accordance with a database of prosodic "portraits".

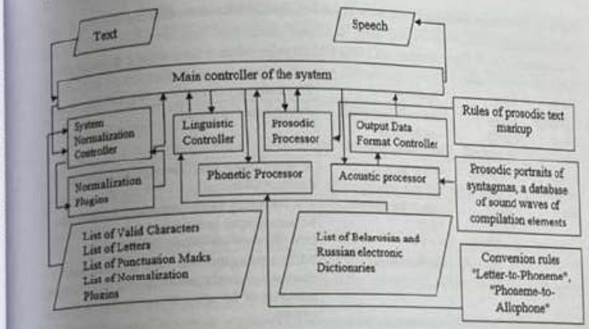

Text
Speech
Main controller of the system
System Normalization Controller
Linguistic Controller
Prosodic Processor
Output Data Format Controller
Rules of prosodic text markup
Normalization Plugins
Phonetic Processor
Acoustic processor
Prosodic portraits of syntagmas, a database of sound waves of compilation elements
List of Valid Characters
List of Letters
List of Punctuation Marks
List of Normalization Plugins
List of Belarusian and Russian electronic Dictionaries
Convenion rules "Letter-to-Phoneme", "Phoneme-to-Allophone"

Fig. 1 The structure of the TTS software implementation

- The phonetic processor converts the text into a phonemic transcriptioa and generates positional and combinatorial allophones. It uses rules to convert the text into a sequence of phonemes (letter-to-phoneme conversion) and rules to convert 392 phoneme sequences into an allophone sequence (phoneme-to-allophone conversion).
- The acoustic processor generates a speech signal by compiling segments of natural sound waves of the corresponding allophones and multiphones. It uses information about which allophones need to be synthesized, as well as which prosodic characteristics should be attributed to each allophone. The text, prosodic, phonetic, and acoustic processors of the speech synthesizer are basically language-independent, and the language specifics (in our case, Belarusian) are set by an appropriate line-up of databases and knowledge, *i.e.*, linguistic rules.

We show the architecture of the TTS system in Fig. 1. Modules that control the sequence of actions of other modules are *controllers*, while modules that implement algorithms for processing a text or speech signal are *processors*. The system's main controller performs the sequence of transformations on the input data, receiving intermediate results at each stage and then transmitting them to the next stage. The text normalization controller removes characters from the text that are unnecessary for speech synthesis, as well as accidental duplications of punctuation marks, standardizes character variants, and removes from the text invalid characters using character replacement rules. The resulting text constitutes the input of the linguistic controller, the output of which is a prosodically marked text and feeds the phonetic processor, which applies "letter-phoneme" and "phoneme-allophone" rule transformations.

The next processing is performed by the prosodic processor, which sets the current values of the amplitude, pitch frequency, and duration of each allophone.

Table 1 Belarusian Dictionary database architecture and CVocReader software

BED database architecture		Architecture of the CVocReader software for reading a dictionary	
Field name	Type	Open/close	The dictionary is switched to read/share mode
Spelling word	String (≤255 characters)	Get the number of words in the dictionary	Real number of words
Stress position	Vector of integers	Word Search	A record (three fields) is selected from the database table
Tag	String (≤255 characters)	Search for the next word	The remaining entries are selected (there may be homographs in the dictionary, so this function will show all homographs stored)
		Get the latest error	While reading the dictionary, this function must be constantly checked for the next correct operation of the entire speech synthesis process

A prosodically marked allophone text is then processed by the acoustic processor, which generates the speech signal, using a handcrafted database of sound waves of allophones and multi-phones. The result of the conversions goes to the output data format controller, which converts it to the sound file in the desired format (wav or *mp3*), as described in Hetsevich and Lobanov (2010).

The text processor performs some preprocessing of the input text (morphological and accent marking of the words), covering 2,097,967 Belarusian wordforms, as in the dictionary (Biryla 1987). The Belarusian Electronic Dictionary contains three types of entries: spelling words, stress positions in words, word tags. The program interface CVocReader is used to manage it. Its architecture is illustrated in Table 1.

For example, when the text processor processes the sentence *Груша цвіла апошні год*..., it produces the following tagging result:

*w гру+ша_НевядомаяКатэгорыя_sbm2012initial_гру+ша_NNIFO_*
sbm1987_гру+ша_NFN1_noun2013_3
w цвіла+_VIIPF_sbm1987_цвіла+_дзеяслоў_verb2013_цвіла+_
_words_processed_3
wапо+шні_НевядомаяКатэгорыя_sbm2012initial_апо+шні_JJMO_sbm1987_а-
*по+шні_JJMA_*
*sbm1987_апо+шні_прыметнік_adjective2013_апо+шні_прыметнік_*
adjective2013_5
*w го+д_НевядомаяКатэгорыя_sbm2012initial_го+д_NNIMO_sbm1987_го+д_*
*NNIMA_sbm1987_*
го+д_NMN1_noun2013_го+д_NMA1_noun2013_5
p.
p newline

The interface of TTS is shown in Fig. 2. To get a synthesized speech, users type in a text in the input field and then click "Generate synthesized speech!". Users can then click "Listen to generated speech" or "Download generated speech file".

To get better results, users can insert the following marks in the input text:

- plus / + / or an accent / а́ / – to define the main stress (for example, "звыча+йны");
- equal to / = / or gravis / э̀ / – to define the secondary stress (for example, "тэ=леперада+ча");
- circumflex / ^ / – between two words, to combine them into a complex phonetic form (for example, "на^стале+", "сказа+ў^бы").

While processing a text, the system produces intermediate results, including a normalized text, a phonemic text, an allophonic text, etc. (Fig. 3). These results may be used to solve other computer-linguistic problems, such as to transcribe the text in Cyrillic, in IPA, or in X-SAMPA.

The "Tokens" window displays five types of characters: alphabet (characters of the target alphabet—the languages selected for synthesis); other letters (not of the target alphabet); digits; whitespaces (whitespace characters: space, line feed, tab, etc.); other characters.

The "Text" window displays data analysis on the wordforms, POS categories, and certain morphological features.

The "Stressed tokens" window displays a list of words with user-placed accents, and "unknown tokens"—the list of words that are not listed in the system database. To determine the position of stress in a wordform, the speech synthesizer checks each word in the input text in dictionaries; indeed, words with the same spelling might have different stresses. The system shows the information about words with an ambiguous accent in the "Homographs" window.

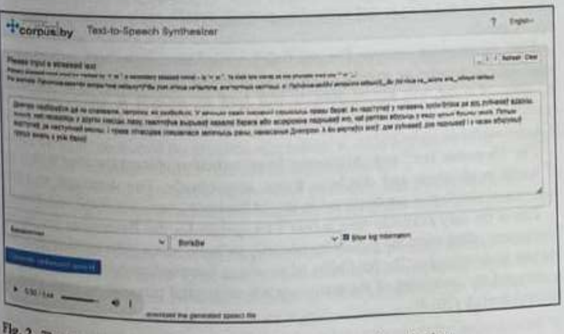
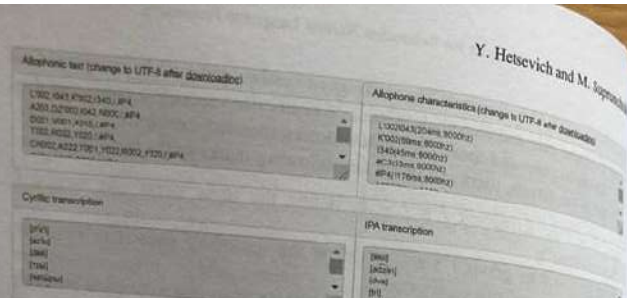


Fig. 2 The interface of the Internet version of the Text-to-speech synthesizer



Allophonic text (change to UTF-8 after downloading)

Allophone characteristics (change to UTF-8 after downloading)

Cyrillic transcription

IPA transcription

Simplified IPA transcription

X-SAMPA transcription

Fig. 3 Intermediate results produced by the Text-to-speech synthesizer

The "Intonation markers" window displays information about intonation markup. Speech synthesis is carried out according to sentences that are characterized by a sufficient degree of intonational autonomy in the text which, in their turn, form separate syntagmas. As a rule, a syntagma consists of one word or a combination of words that have a certain semantic and intonational completeness. Due to the complexity and insufficient development of rules for extracting syntagmas, it is only possible to perform a superficial syntactic analysis using available morphosyntactic information about the phrases and punctuation. To automatically produce the text-to-speech conversion, we propose to determine the intonation types of the syntagmas in narrative, interrogative, and exclamative sentences, according to the formal markers presented in the first table. For each punctuation sign, we compute its formal marker and intonation portrait, which replaces the punctuation sign (Table 2).

Thus, the narrative sentence *Стары бабёр з палёгкаю ўздыхнуў – след вады не абмане, прывядзе да вады! (Алесь Жук)* [The old Beaver sighed with relief—the trace of water will not deceive, it will lead to water!] is analyzed by the synthesizer as: *Стары бабёр з палёгкаю ўздыхнуў (C4) след вады не абмане (C3) прывядзе да вады (E2).*

The "Phonemic Text" and "Allophonic Text" windows present the user with a list of words in phonemic and allophone forms, respectively. The durations and frequencies of allophones are displayed in the "Allophone characteristics" window.

This is the only available open access TTS system for the Belarusian language. The system offers users the possibility not only to listen to a synthesized audio file, but also to download it. The possibility of accessing intermediate results allows us to understand the intricacies of the text-to-speech sequential process, as presented in Hetsevich et al. (2019).

Table 2 Correspondence type of intonation-formal marker

Type of utterance	Type of intonation	Punctuation sign and an intonation portrait
Narrative	Finality	P1—intonation of "colon"—[:], P2—intonation of "introduction"—[ )], P3—intonation of "semicolon"—[;], P4—intonation of "dot"—[.], P5—intonation of "ellipsis"—[...], P6—intonation of "paragraph"—[#]
	Non-finality	C1—intonation of "conjunction AND", C2—intonation of "conjunction OR", C3—intonation of "comma"—[,], C4—intonation of "dash"—[–], C5—intonation of "pre-introduction"—[(], C6—intonation of lexical syntagmas
Question	Interrogative	Q1—single-syntagma question with a question word, Q1-1—a question with a question word, containing two or more syntagmas, Q2—single-syntagma question without a question word, Q2-1—a question without a question word, containing two or more syntagmas
Exclamation	Exclamative/ imperative	E1—single-syntagma exclamation with an interjection, E1-1—an exclamation with an interjection, containing two or more syntagmas, E2—single-syntagma exclamation, E2-1—an exclamation, containing two or more syntagmas

3 The Transcription Generator

The Transcription Generator is the most illustrative proof of the importance of linguistic rule-based approach in NLP. The service converts graphical (alphabetical) Belarusian wordforms into a phonetic transcription, following the norms of modern pronunciation.

TTS systems necessarily include several transcription generation algorithms. Such systems usually contain several processors for each stage of the transcription, as we have seen in Fig. 1. An incorrect result produced at a stage by one of the processors significantly deteriorates the final result therefore it is crucial to be able to detect mistake produced by process, and correct it. This is possible if all processes are based on handcrafted data that can be easily corrected.

The "grapheme-to-phoneme" conversion process is described in Hetsevich et al. (2014). Its algorithm determines the sequence of phonemes that correspond to the input text.

In Belarusian, many word spellings are close to their pronunciation and can thus be described by rules; our system uses these rules to convert graphic forms to their phonetic representation. However, it is not possible to produce the phonetic transcription of a wordform directly from its graphical form, we should use an intermediate level to represent allophones. This intermediate representation is constituted by a sequence of allophones and pauses separated by commas, cf. (Zahariev et al. 2016), as shown in the following:

- Original text: *Напэўна не скажу, над якою рэчкаю векаваў стары дуб; ці то над Нёманам, ці то над Свіслаччу* (Якуб Колас).
- Translation: "I don't think I can say for certain beside which river the old oak-tree stood,—whether it was the river Nioman, or whether it was the Śvislač" (Yakub Kolas);
- TTS format: >,N004,A221,>,P001,E011,W013,>,N004,A223,/,>, N002, E042,/,>,S002,K004,A232,>,ZH002,U020,/,>,#C3, >,N002,A022,T002,/,>, J'002,A242,>,K001,O033,>,J'012,U242,/,>, R002,E022,>,CH002,K004, A333,>,J'012,U343,/,>,V'012,E342,>,K004,A231,>,V011,A011,W013,/,>, S002,T002,A222,>,R002,Y022,/,>,D002,U021,P000,/,>,#P1, >,C'001,I042,/, >,T001,O022,/,>,N002,A022,T002,/,>,N'002,O141,>,M002,A112,>,N002, A121,M000,/,>,#C3, >,C'001,I042,/,>,T001,O022,/,>, N002,A022,T002,/,>, S'002,V'001,I042,>,S002,L004,A312,>,CH102,U320,/,>,#P4,
- Cyrillic transcription: [напэ́ўна] [н'э] [скажу́] | [на́т] [йако́йу] [рэ́чкайу] [в'экава́ў] [стары́] [ду́п] ‖ [ц'і́] [то́] [на́т] [н'о́манам] | [ц'і́] [то́] [на́т] [с'в'і́слач:у] ‖
- International Phonetic Alphabet transcription: [naˈpɛwna] [ˈnʲɛ] [skaˈʐu] | [ˈnat] [jaˈkɔju] [ˈre□kaju] [vʲekaˈvaw] [staˈri] [ˈdup] ‖ [ˈtsʲi] [ˈtɔ] [ˈnat] [ˌnʲɔˌmaˌnam] | [ˈtsʲi] [ˈtɔ] [ˈnat] [ˈsʲvʲisla□□u] ‖

To produce the final transcription, it is necessary to access a database of correspon- dences of the form allophone⇔transcription. Each allophone is represented by a code constituted by one, two or three Latin letters, an apostrophe sign, and three Arabic numerals that characterize the type of junction of the phoneme. The initial allophone database contained 960 different allophones that were described manu- ally. An analysis of the correspondence between allophones and their phonetic transcription showed that abbreviated allophone designations, especially phoneme names, signs of softness, and first index are sufficient for the transcription. Thanks to this observation, we were able to decrease the number of allophone⇔transcription correspondences to 99. Linguists developed this list of correspondences, following the guide (Padluzhny 1989). We give a fragment of the resulting list in Table 3.

Thus, the transcription of Belarusian texts uses the following resources:

- the set of correspondences "punctuation mark—intonation mark $P = << p_1,$ $int_1 >, \ldots, < p_k, int_k >>$, where p_k—k-i is the punctuation mark, int_k—k-i is the intonation mark, k is the number of correspondences;
- a database of "grapheme-phoneme" conversion rules;

Table 3 Fragment of the list of correspondences "allophone-transcription"

Abbreviated allophone code	A0	A1	A2	B0	B'0	B1	B'1
Transcription	а́	Ʌ	ᵃ	б	б'	Б:	б'

- a database of "phoneme-allophone" conversion rules;
- a database of correspondence "allophone-transcription".

For example, the most common approach performed during the "grapheme-to-phoneme" conversion is to process character sequences from left to right. For each character of the input, we apply one or more rules to generate its corresponding phoneme. The presence of two assimilation effects of the previous consonant phoneme is typical in Belarusian: deafness-sonority, hardness-softness. Moreover, it is necessary to note that the effect of assimilation on deafness-sonority can be intra-word and inter-word. At the same time, their distribution to neighboring graphemes goes from right to left. Since the indicated effects do not affect each other, it is possible to process the transformation "grapheme-phoneme" in four consecutive stages:

1. Verify that the grapheme complies with the rules that treat canonical changes, check for the effects of assimilation of consonant phonemes by deafness-sonority, and replace it (in case of coincidence) with the corresponding phoneme group.
2. Replace a letter with a phoneme according to standard rules.
3. Check the softness of the previous grapheme (a necessary but insufficient condition for softness).
4. Check the grapheme for compliance with the softening rules, and add softness to this phoneme in case of a match.

The structure of expert rules consists of four blocks:

1. "Standard" rules replace a grapheme with a phoneme. For example, the grapheme "A" in Belarusian is often replaced with the phoneme "A".
2. Exceptions to the standard replacement rules are expressed with regular expressions. They can represent an assimilation effect, a replacement, etc. For example, in Belarusian, the grapheme "Г" can turn into phonemes "G" or "GH", depending on its right context: *горка* [slide; hill]—*GH,O,+,R,K,A*; *гузік* [button]—*G,U,+,Z',I,K*.
3. Softening graphemes, *i.e.*, those graphemes before which the soft sound may appear.
4. Softening rules expressed as regular expressions. We have described the condition under which a grapheme will turn into a soft phoneme. For example, grapheme *Н* turns into soft phoneme *Н'* before the sequence of consonants *ДЗ, Ц, Й* and vowels *Е, Ё, Ю, Я, І, Ь* or *С, Л, Ц, З*.

Examples of formalized grapheme-phoneme conversion rules are presented in Table 4.

The Transcription Generator provides an accurate result in more than 98% of the cases. It has significantly facilitated the construction of the Orthoepic Dictionary of the Belarusian Language (Rusak 2017), which contains 117,000 headings, as well as different implementations of the dictionary (Fig. 4).

Table 4 Fragment of "grapheme-phoneme" conversion rules for the Belarusian language

Standard "grapheme-to-phoneme" rules	Exceptions to the standard rules	Softening graphemes	Standard softening rules
Ж–ZH	Д(С)ТВ–С	Е	(Н)[ДЗЦЙ][ЕЁЮЯІЬ]
З–Z	(Д)[КСПТФХЦЧШ]–Т	Ё	(Н)[СЛЦЗ]
І–І	(Т)[БГДЗЖ]–D	Ю	(Л)[Л]
Й–J'	(З)ДЖ–ZH	Я	(М)[М]
К–К	(З)[КПСТФХЦЧШ]–S	І	([ЗСН])[Д]

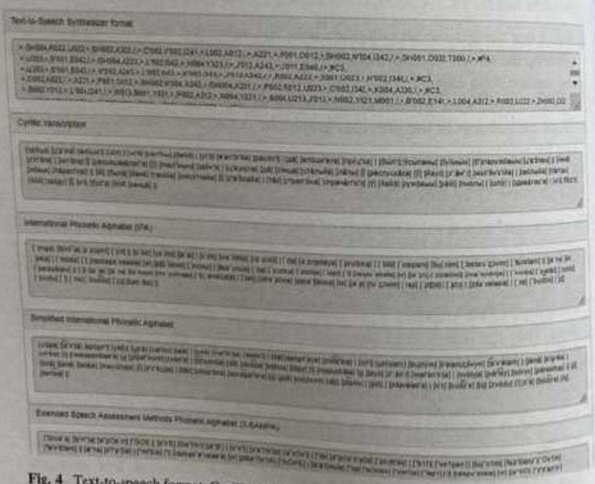

Text-to-Speech Synthesizer format
Cyrillic transcription
International Phonetic Alphabet (IPA)
Simplified International Phonetic Alphabet
Extended Speech Assessment Methods Phonetic Alphabet (X-SAMPA)

Fig. 4 Text-to-speech format, Cyrillic, IPA and X-Sampa outputs

4 Word Paradigm Generator

The "Word Paradigm Generator" service is free and available at https://corpus.by/WordParadigmGenerator. It receives a word as input and returns its corresponding paradigm. If the word is not listed in the dictionary, the service returns a paradigm associated with a word similar to the word in the input; for more details see (Hetsevich et al. 2016).

This service uses the grammatical dictionary of the Belarusian language (https://corpus.by/VoicedElectronicGrammaticalDictionary/?lang=be), which covers a large vocabulary, but has certain gaps, primarily in technical terminology. Therefore, it is necessary to update it. We have used this service to compile terminological dictionaries in the legal domain as well as in the medical domain, as presented in Varanovich et al. (2021).

Texts that need to be processed often contain words unknown to the system. When voicing a text, it is very important to determine the stress in each word because the stress in Belarusian is unfixed and mobile.

The main goal of this service is not only to compute the category, but also the whole paradigm. The algorithm our team designed for the automatic generation of word paradigms consists of 11 consecutive interdependent steps, which produce the most suitable paradigms for a given word. The algorithm looks for the closest paradigm(s) for the last letters of a word, and is presented in flow graph 1 (Fig. 5).

The dictionary used by the automatic synthesizer might contain mistakes in setting the location of the stress. Therefore, it must be updated when the system encounters words unknown to the automatic speech synthesizer or when the services "Spell checker" and "Voiced Electronic Grammatical Dictionary" produce new words.

The service "Unknown Words Processor" (https://corpus.by/UnknownWordsProcessor) presents the list of unknown words and allows users to set their stress and their POS category. However, this service does not consider the lemma and other variants of the wordform (for example, its singular and plural forms in six cases for nouns).

The "Word Paradigm Generator" service is used to completely close the gaps in the dictionary. It sets not only the POS category of the word, but also produces one or several most suitable paradigms for the word, using the last letters of a word to get its most likely paradigms. The algorithm consists of 11 consecutive interdependent steps. Figure 6 displays the graphical interface of the service.

The interface contains the following areas:

- input field for the wordform(s);
- choice of the dictionary (about NooJ format see (Silberztein 2003));
- optional selection of a tag and/or a POS category;
- the button "Generate possible paradigms!" which starts the processing and returns the results.

Tags displayed after the "_" symbol show the grammatical meaning of the word, for example, POS category, gender, number, case, etc., and generate paradigms of the input word based on similar words with the same grammatical meaning if the input word cannot be found in the service dictionaries, as described in Zanouka (2017).

The system can generate the paradigms for the wordform if it is not described in the dictionary, using words. For example, there is no paradigm of the word *аўдыягід* in the dictionary. When the service is executed, it will produce paradigms based on all words similar in writing and found in the dictionary—*гід*, *альдэгід*, *поліфармальдэгід*, *фармальдэгід*, *агід*, *эгід*, see Fig. 7.

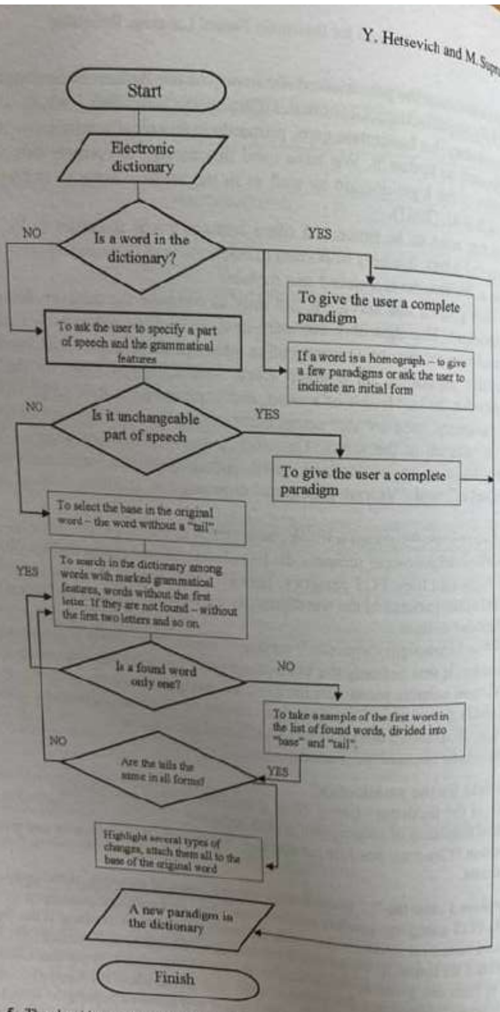
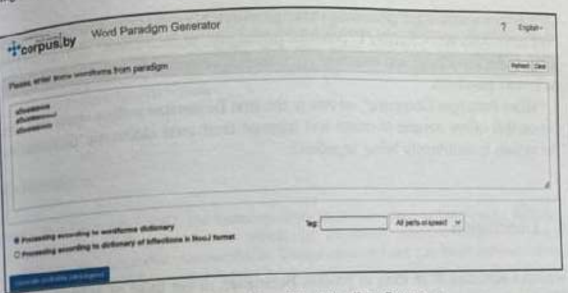

Start
Electronic dictionary
Is a word in the dictionary?
NO
YES
To give the user a complete paradigm
If a word is a homograph – to give a few paradigms or ask the user to indicate an initial form
To ask the user to specify a part of speech and the grammatical features
Is it unchangeable part of speech
NO
YES
To give the user a complete paradigm
To select the base in the original word – the word without a "tail".
To search in the dictionary among words with marked grammatical features, words without the first letter. If they are not found – without the first two letters and so on.
YES
Is a found word only one?
NO
To take a sample of the first word in the list of found words, divided into "base" and "tail".
NO
Are the tails the same in all forms?
YES
Highlight several types of changes, attach them all to the base of the original word
A new paradigm in the dictionary
Finish

Fig. 5 The algorithm of the service "Word Paradigm Generator"

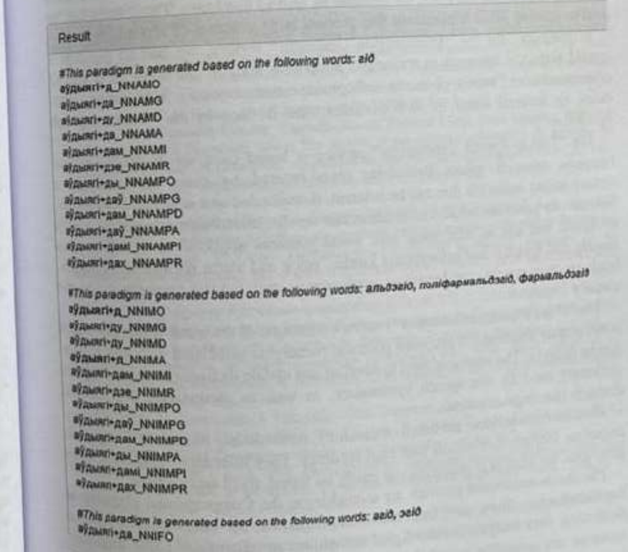


Please enter some wordforms from paradigm

● Processing according to wordforms dictionary
○ Processing according to dictionary of inflections in [illegible] format

Tag:
All parts of speech

Fig. 6 The graphical interface of the service «Word Paradigm Generator»

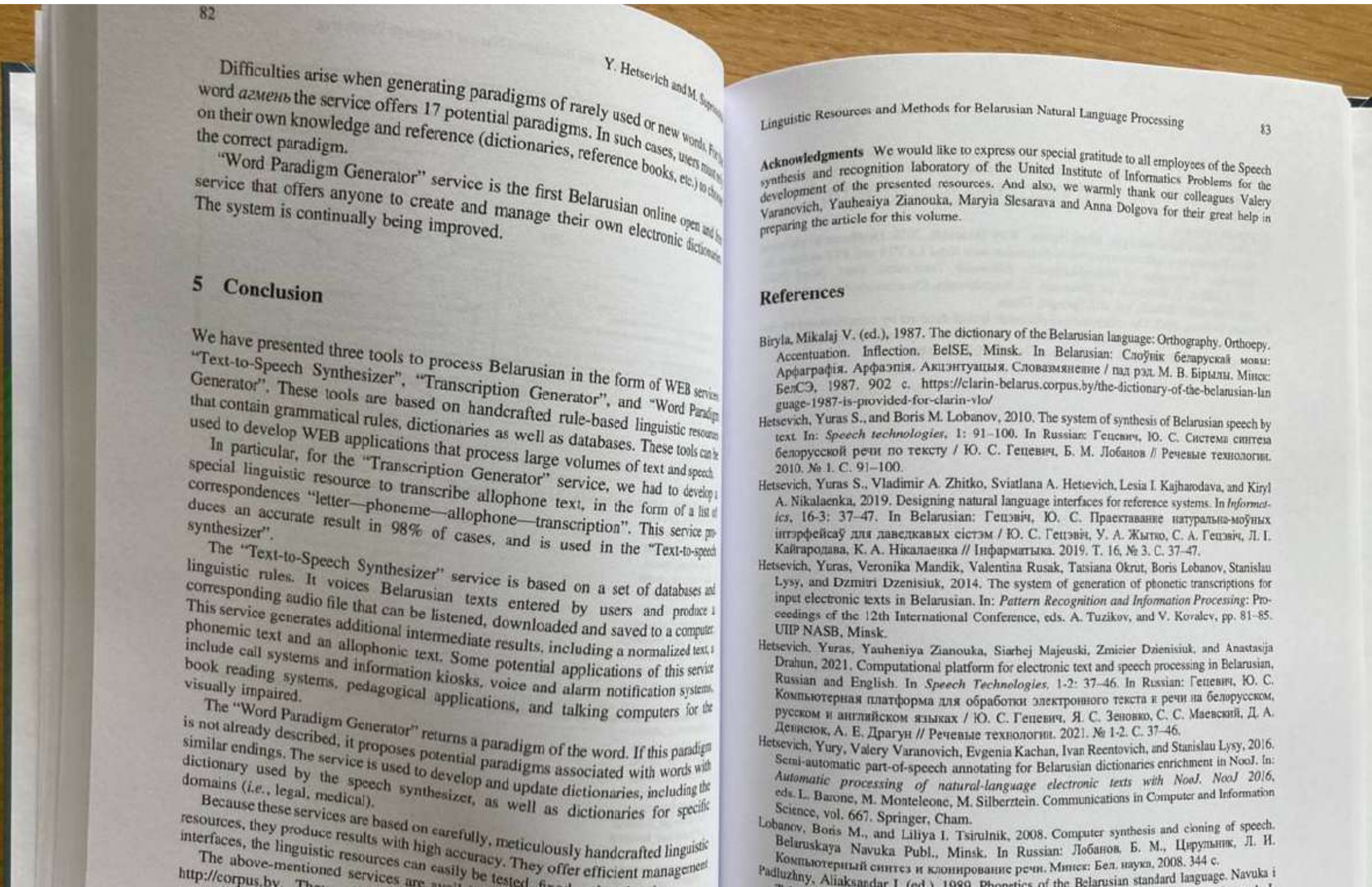

```
Result

#This paradigm is generated based on the following words: гід
аўдыягі+д_NNAMO
аўдыягі+да_NNAMG
аўдыягі+ду_NNAMD
аўдыягі+да_NNAMA
аўдыягі+дам_NNAMI
аўдыягі+дзе_NNAMR
аўдыягі+ды_NNAMPO
аўдыягі+даў_NNAMPG
аўдыягі+дам_NNAMPD
аўдыягі+даў_NNAMPA
аўдыягі+дамі_NNAMPI
аўдыягі+дах_NNAMPR

#This paradigm is generated based on the following words: альдэгід, поліфармальдэгід, фармальдэгід
аўдыягі+д_NNIMO
аўдыягі+ду_NNIMG
аўдыягі+ду_NNIMD
аўдыягі+д_NNIMA
аўдыягі+дам_NNIMI
аўдыягі+дзе_NNIMR
аўдыягі+ды_NNIMPO
аўдыягі+даў_NNIMPG
аўдыягі+дам_NNIMPD
аўдыягі+ды_NNIMPA
аўдыягі+дамі_NNIMPI
аўдыягі+дах_NNIMPR

#This paradigm is generated based on the following words: [illegible], эгід
аўдыягі+да_NNIFO
```

Fig. 7 Potential paradigms generated for a missing word



Difficulties arise when generating paradigms of rarely used or new words. Fo word *агмень* the service offers 17 potential paradigms. In such cases, users must on their own knowledge and reference (dictionaries, reference books, etc.) to cho the correct paradigm.

"Word Paradigm Generator" service is the first Belarusian online open and f service that offers anyone to create and manage their own electronic dictionaries. The system is continually being improved.

5 Conclusion

We have presented three tools to process Belarusian in the form of WEB services "Text-to-Speech Synthesizer", "Transcription Generator", and "Word Paradigm Generator". These tools are based on handcrafted rule-based linguistic resources that contain grammatical rules, dictionaries as well as databases. These tools can be used to develop WEB applications that process large volumes of text and speech.

In particular, for the "Transcription Generator" service, we had to develop a special linguistic resource to transcribe allophone text, in the form of a list of correspondences "letter—phoneme—allophone—transcription". This service produces an accurate result in 98% of cases, and is used in the "Text-to-speech synthesizer".

The "Text-to-Speech Synthesizer" service is based on a set of databases and linguistic rules. It voices Belarusian texts entered by users and produce a corresponding audio file that can be listened, downloaded and saved to a computer. This service generates additional intermediate results, including a normalized text, a phonemic text and an allophonic text. Some potential applications of this service include call systems and information kiosks, voice and alarm notification systems, book reading systems, pedagogical applications, and talking computers for the visually impaired.

The "Word Paradigm Generator" returns a paradigm of the word. If this paradigm is not already described, it proposes potential paradigms associated with words with similar endings. The service is used to develop and update dictionaries, including the dictionary used by the speech synthesizer, as well as dictionaries for specific domains (*i.e.*, legal, medical).

Because these services are based on carefully, meticulously handcrafted linguistic resources, they produce results with high accuracy. They offer efficient management interfaces, the linguistic resources can easily be tested, fixed and updated.

The above-mentioned services are available on the Computational platform at http://corpus.by. They are free and complemented by courses in language processing, data analysis for the digital humanities in Belarusian. These linguistic resources are available to scholars, researchers and scientists from all spheres through single sign-on access.

Acknowledgments We would like to express our special gratitude to all employees of the Speech synthesis and recognition laboratory of the United Institute of Informatics Problems for the development of the presented resources. And also, we warmly thank our colleagues Valery Varanovich, Yauheniya Zianouka, Maryia Slesarava and Anna Dolgova for their great help in preparing the article for this volume.

References

Biryla, Mikalaj V. (ed.), 1987. The dictionary of the Belarusian language: Orthography. Orthoepy. Accentuation. Inflection. BelSE, Minsk. In Belarusian: Слоўнік беларускай мовы: Арфаграфія. Арфаэпія. Акцэнтуацыя. Словазмяненне / пад рэд. М. В. Бірылы. Мінск: БелСЭ, 1987. 902 с. https://clarin-belarus.corpus.by/the-dictionary-of-the-belarusian-language-1987-is-provided-for-clarin-vlo/

Hetsevich, Yuras S., and Boris M. Lobanov, 2010. The system of synthesis of Belarusian speech by text. In: *Speech technologies*, 1: 91–100. In Russian: Гецевич, Ю. С. Система синтеза белорусской речи по тексту / Ю. С. Гецевич, Б. М. Лобанов // Речевые технологии. 2010. № 1. С. 91–100.

Hetsevich, Yuras S., Vladimir A. Zhitko, Sviatlana A. Hetsevich, Lesia I. Kajharodava, and Kiryl A. Nikalaenka, 2019. Designing natural language interfaces for reference systems. In *Informatics*, 16-3: 37–47. In Belarusian: Гецэвіч, Ю. С. Праектаванне натуральна-моўных інтэрфейсаў для даведкавых сістэм / Ю. С. Гецэвіч, У. А. Жытко, С. А. Гецэвіч, Л. І. Кайгародава, К. А. Нікалаенка // Інфарматыка. 2019. Т. 16, № 3. С. 37–47.

Hetsevich, Yuras, Veronika Mandik, Valentina Rusak, Tatsiana Okrut, Boris Lobanov, Stanislau Lysy, and Dzmitri Dzenisiuk, 2014. The system of generation of phonetic transcriptions for input electronic texts in Belarusian. In: *Pattern Recognition and Infomation Processing*: Proceedings of the 12th International Conference, eds. A. Tuzikov, and V. Kovalev, pp. 81–85. UIIP NASB, Minsk.

Hetsevich, Yuras, Yauheniya Zianouka, Siarhej Majeuski, Zmicier Dzienisiuk, and Anastasija Drahun, 2021. Computational platform for electronic text and speech processing in Belarusian, Russian and English. In *Speech Technologies*, 1-2: 37–46. In Russian: Гецевич, Ю. С. Компьютерная платформа для обработки электронного текста и речи на белорусском, русском и английском языках / Ю. С. Гецевич, Я. С. Зеновко, С. С. Маевский, Д. А. Денисюк, А. Е. Драгун // Речевые технологии. 2021. № 1-2. С. 37–46.

Hetsevich, Yury, Valery Varanovich, Evgenia Kachan, Ivan Reentovich, and Stanislau Lysy, 2016. Semi-automatic part-of-speech annotating for Belarusian dictionaries enrichment in NooJ. In: *Automatic processing of natural-language electronic texts with NooJ. NooJ 2016*, eds. L. Barone, M. Monteleone, M. Silberztein. Communications in Computer and Information Science, vol. 667. Springer, Cham.

Lobanov, Boris M., and Liliya I. Tsirulnik, 2008. Computer synthesis and cloning of speech. Belaruskaya Navuka Publ., Minsk. In Russian: Лобанов, Б. М., Цирульник, Л. И. Компьютерный синтез и клонирование речи. Минск: Бел. наука, 2008. 344 с.

Padluzhny, Aliaksandar I. (ed.), 1989. Phonetics of the Belarusian standard language. Navuka i Tehnika Publ., Minsk. In Belarusian: Фанетыка беларускай літаратурнай мовы / рэд. А. І. Падлужны. Мінск: Навука і тэхніка, 1989. 335 с.

Rusak, Valentina P. (ed.), 2017. Orthoepical dictionary of the Belarusian language. Bielaruskaja Navuka Publ., Minsk. In Belarusian: Арфаэпічны слоўнік беларускай мовы / рэд. В. П. Русак. Мінск: Бел. навука, 2017. 757 с.

Silberztein, Max 2003: NooJ Manual. https://nooj.univ-fcomte.fr/downloads.html.

Taylor, Paul, 2009. Text-to-Speech synthesis. Cambridge University Press, N. Y.

Varanovich, Valery, Mikita Suprunchuk, Yauheniya Zianouka, Tsimafei Prakapenka, Anna Dolgova, and Yuras Hetsevich, 2021. Creation of a legal domain corpus for the Belarusian module in NooJ: Texts, Dictionaries, Grammars. In: *15th International Conference NooJ 2021:* book of abstracts, eds M. Bigey, A. Richton, M. Silberztein, I. Thomas: 36–37. Besançon, France.

Zahariev, Vadim, Stanislau Lysy, Alena Hiuntar, Yury Hetsevich, 2016: Grapheme-to-phoneme and phoneme-to-grapheme conversion in Belarusian with NooJ for TTS and STT systems. In: *Automatic Processing of Natural-Language Electronic Texts with NooJ. NooJ 2015*, eds. T. Okrut, Y. Hetsevich, M. Silberztein, H. Stanislavenka. Communications in Computer and Information Science, vol. 607. Springer, Cham.

Zanouka, Evgenia, 2017. The enlargement of electronic lexical database by computational on-line free system. In: *Open Semantic Technologies for Intelligent Systems*: Proceedings of the International Conference, ed. V. Golenkov: 179–182. BSUIR, Minsk.