Grapheme-to-Phoneme and Phoneme-to-Grapheme Conversion in Belarusian with NooJ for TTS and STT Systems

Vadim Zahariev[1(✉)], Stanislau Lysy[2], Alena Hiuntar[2], and Yury Hetsevich[2]

[1] The Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
zahariev@bsuir.by

[2] The United Institute of Informatics Problems of National Academy of Sciences of Belarus, Minsk, Belarus
stanislau.lysy@gmail.com, lena205593@gmail.com,
yury.hetsevich@gmail.com

Abstract. To process texts in any language, a full and thorough description of the given language is required. The authors of this article have noted that, while much has been done in the development of language processing with NooJ, so far little attention has been paid to issues related to phonetic language features.

Keywords: Text-to-speech · Speech-to-text · Grapheme-to-phoneme conversion · Phoneme-to-grapheme conversion · The Belarusian language · Phonetics · Transcription

1 Introduction

Natural language processing (NLP) is a field of computer science closely related to linguistics, and plays a significant role in advancing communication between human and computer. It is needed to transform relevant information locked in text into structured data that can be used by computer processes aimed at improving various aspects of life.

There are two main problems in the transition step between morphological and phonetic levels of texts. The first one is a transformation of orthographic text into its phonetic representation for further processing in the phonetic encoding/decoding step. The task of this transformation is very common in text-to-speech (TTS) systems (Fig. 1, solid lines). The second one is an inverse problem: the building of written orthographic text from transcribed spoken language. This issue is an important aspect within the framework of speech-to-text (STT) systems (Fig. 1, dashed lines). In both cases of grapheme-to-phoneme (G2P) and phoneme-to-grapheme (P2G) conversion, it is difficult to develop separate statistical models for all phonemes and words in TTS and STT systems with a rather large vocabulary. This is especially true for relatively localized languages, like Belarusian, when it is hard to collect a great amount of speech data for statistical models in training procedures.

T. Okrut et al. (Eds.): NooJ 2015, CCIS 607, pp. 137–150, 2016.
DOI: 10.1007/978-3-319-42471-2_12

We consider NooJ [1] to be an effective tool in solving these problems, as it can be used for creating dictionaries of correspondences between orthographic words and their phonetic transcriptions, as well as for development of rule-based grammars for G2P and P2G conversions.

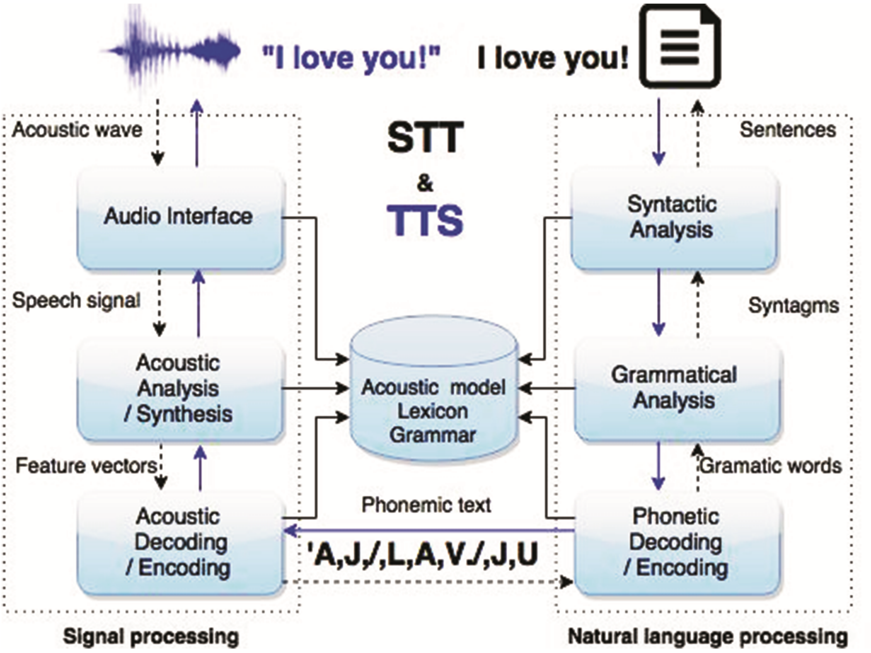


Fig. 1. The general scheme of the STT and the TTS systems

2 Specific Features of Belarusian in the Context of G2P and P2G Conversion

For both G2P and P2G conversion processes, a fundamental aspect is the choice of approach to the problem. There are two common approaches to the problem of G2P and P2G conversions: the first is related to statistical methods and uses phonetically transcribed corpuses or pronunciation dictionaries; the second is a rule-based method. The choice of approach greatly depends on the type of orthography of the processed language. An orthography in which the correspondences between spelling and pronunciation are highly complex or inconsistent is called a deep orthography. In such a case, statistical methods, or special databases, are used. Another type of orthography is called shallow orthography, and it is determined by relatively simple and consistent correspondences between spelling and pronunciation. In this case, the rule-based methods are more preferable.

Regarding the Belarusian language, we should take into account the fact that its orthography is based on a simple phonemic principle—words are pronounced as they are spelled. This is particularly true for Belarusian Classical Orthography, or *Taraškievica,* where the written and the phonetic forms are almost identical. In modern Belarusian, this is not so evident, but it is true to a greater degree than, for instance, in Russian. In Table 1, a short comparison of correspondences between some orthographic words and their transcriptions for Belarusian and Russian is presented [2].

Table 1. A comparison of correspondences between orthographic words and their phonetic transcriptions in Belarusian and Russian

English word	Belarusian/ Russian	Orthographic word	Cyrillic transcription	IPA transcription
River	Belarusian	рэчка	[*рэ́чка*]	[*'rɛʧka*]
	Russian	речка	[*р'э́ч'ка*]	[*'rɛʧ'ka*]
Sun	Belarusian	сонца	[*со́нца*]	[*'sɔnʦa*]
	Russian	солнце	[*со́нцэ*]	[*'sɔnʦɛ*]
Laugh	Belarusian	смяяцца	[*с'м'айа́цца*]	[*s^{j}m^{j}a'jaʦʦa*]
	Russian	смеяться	[*см'ийа́цца*]	[*sm^{j}i'jaʦʦa*]

Due to the relative proximity of phonetic forms and spelling of the words in the Belarusian language, the G2P and the P2G conversion algorithms based on the rules are more preferable. Nevertheless, there are a number of Belarusian language features that create difficulties in performing these two kinds of transformations. Let us take a brief look at some examples.

The Belarusian language has ten letters representing vowel sounds. They can be divided into two categories: non-iotified (***а, о, у, э, і, ы***) and iotified (***я, ё, ю, е***) vowels. Letters in the second sequence represent four sounds of the first sequence, but with an initial [*j*] sound. When a consonant precedes an iotified vowel, it becomes palatized. Taking into account this fact, we have to define a rule that makes a consonant preceding an iotified vowel palatized for the G2P conversion, and in P2G conversion a rule that converts a non-iotified vowel into an iotified analogue after palatized consonant is needed.

The correspondences between orthography and phonetics in Belarusian also depend on accent position. For instance, vowel ***ё*** is almost always stressed, and when unstressed, ***ё*** changes to ***е*** or ***я*** depending on the accent position, and this is not always transmitted in writing [3].

It is also necessary to mention some of the most common sound changes in Belarusian: assimilation, dissimilation, and accommodation. These features are significant both for G2P and P2G conversations. For example, in the word ***стужка*** (Eng. 'ribbon'), the letter ***ж*** that usually designates the sound [ʐ] changes into the sound [ʂ] as a result of assimilation.

The great value in the P2G conversion plays phenomenon of interaction between sounds existing in the flow of speech, and influence of sounds on each other, causing specific phonetic changes. The Belarusian language is characterized by changing of phonemes under the influence of neighboring elements (combinational), as well as changes resulting from the ratio of phonemes to verbal blows and their absolute location relative to the beginning or the end of the word (positional) [4].

Also almost every rule of Belarusian have its own set of exceptions. Most of them are connected with loanword. This fact should be taken in account while developing algorithms of grapheme-to-phoneme and phoneme-to-grapheme conversion.

3 Grapheme-to-Phoneme Conversion for the Belarusian NooJ Module

Grapheme-to-phoneme (G2P) conversion, or phonetization, is a process of defining the sequence of phonemes required to pronounce a word, phrase, or even a text. The G2P algorithms are used to generate the most probable phoneme list for a word not contained in the pronunciation dictionary. These algorithms are widely used in automated text-to-speech systems.

We developed two ways of obtaining phonetic transcriptions for each word in a Belarusian text. The first way involves creating a Belarusian dictionary in NooJ format, containing information on the pronunciation for each word. The second way involves developing morphological NooJ grammars for generating phonetic transcriptions for orthographic words. As such, we obtain two kinds of phonetic level representation, thus allowing us to perform G2P conversion.

To create the dictionary, a software tool which allows to quickly convert both single words and whole texts into phonetic transcription was developed. This tool is called "Transcription Generator" and is now available as web-service at [5].

The transcription generation system converts a Belarusian text into its phonetic transcription. It takes as an input any orthographic text in Belarusian, with labels of main and side accents. The character '+' (plus sign) after a vowel can be used to mark the main accent and the character '=' (equals sign) – to mark the side accent. As well as these characters, standard accent characters can be used. The system currently supports four types of transcription:

- Cyrillic transcription (based on [6]);
- transcription based on the work by U.A. Koshchanka [7];
- International Phonetic Alphabet (or IPA) [8];
- Extended Speech Assessment Methods Phonetic Alphabet (or X–SAMPA) [9].

The "Transcription Generator" algorithm was developed on the base of the multi-voice TTS system for the Belarusian and Russian languages [10]. The phonetic processor of this system uses two sets of rules. These rules represent almost a full set of grapheme-to-phoneme transformations for Belarusian. The first set of rules is a set of general rules that shows direct correlations between the letter and the phoneme, and the second set shows more complex rules, or exceptions to general rules, depending on letter context. An excerpt of these rules list is given in the Table 2.

Algorithms of transcription generation system convert orthographic texts into the phonemic form developed for the Belarusian TTS system. This phonemic notation could be converted into any common transcription notation. To do this, we created a base of "phoneme – transcription" correlations. An excerpt of this base is presented in Table 3.

Table 2. An excerpt of the list of general rules and their exceptions

General rules	Exceptions to general rules
А-A	(З)[ДГ]V-S
Б-B	(З)ДЖ-ZH
В-V	(З)[КПСТФХЦЧШ]-S
Г-GH	(З)[ЬЪ_V][КПСТФШ]-S
Д-D	(С)[БГДЗЖ]-Z
Е-E	(С)[ЬЪ_V][БВГДЗЖ]-Z
Ё-O	(Ж)[КПСТФХЦЧШ]-SH
…	…

Table 3. An excerpt of list of “phoneme – transcription” correspondences

TTS	Cyrillic	Latin	IPA	X-SAMPA
V0	в	v	v	v
V1	в:	v:	vv	vv
V′1	в′:	v′:	v^{j}v^{j}	v′v′
V′0	в′	v′	v^{j}	v′
GH0	γ	ɣ	ɣ	G
GH1	γ:	ɣ:	ɣɣ	GG
GH′1	γ′:	ɣ′:	ɣ^{j}ɣ^{j}	G′G′
GH′0	γ′	ɣ′	ɣ^{j}	G′
G0	г	g	ɡ	g
G′0	г′	g′	ɡ^{j}	g′
…	…	…	…	…

With the help of the transcription generation system, a phonetic dictionary of arbitrary format can be generated, including a format supported by NooJ. To create a NooJ dictionary with phonetic transcriptions, the following format was developed:

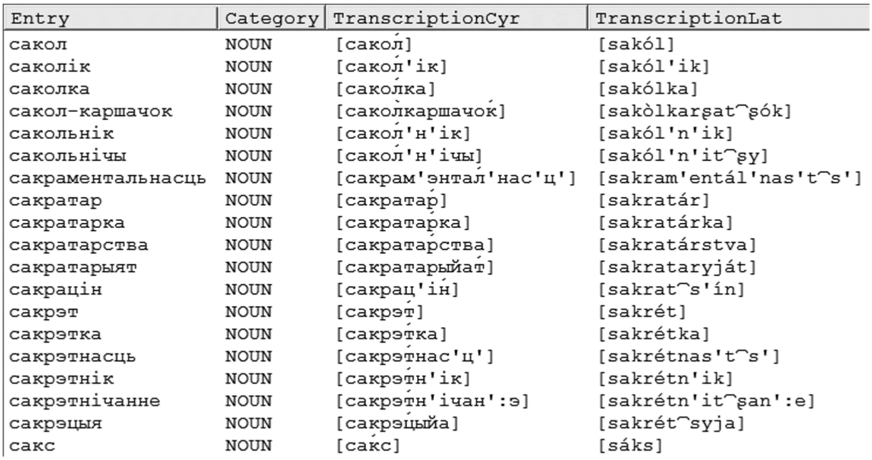
word, *PART–OF–SPEECH*
+TranscriptionCyr = [*transcription*]
+TranscriptionLat = [*transcription*]
+TranscriptionIPA = [*transcription*]
+TranscriptionXSAMPA = [*transcription*].

For instance, the word ***сакаляня*** would have the following view:

сакаляня, NOUN
+TranscriptionCyr = [сакал′ан′á]
+TranscriptionLat = [sakal′an′á]
+TranscriptionIPA = [sakal^{j}a′n^{j}a]
+TranscriptionXSAMPA = [sakal′a″n′a].

To generate a dictionary in NooJ format for all Belarusian words, a special algorithm was developed and implemented. It takes as an input an orthographic dictionary of the

Belarusian language, allocates all dictionary entries, obtains the required information on every entry (word with accent, part-of-speech), uses the transcription generation system to produce phonetic transcriptions of every word in four forms, and compiles all this information into NooJ dictionary format. In the Fig. 2 a NooJ dictionary excerpt is presented.



Entry	Category	TranscriptionCyr	TranscriptionLat
сакол	NOUN	[сакол́]	[sakól]
саколік	NOUN	[сакол́'ік]	[sakól'ik]
саколка	NOUN	[сакол́ка]	[sakólka]
сакол-каршачок	NOUN	[сакол̀каршачоќ]	[sakòlkarʂat͡ʂók]
сакольнік	NOUN	[сакол́'н'ік]	[sakól'n'ik]
сакольнічы	NOUN	[сакол́'н'ічы]	[sakól'n'it͡ʂy]
сакраментальнасць	NOUN	[сакрам'энтал́'нас'ц']	[sakram'entál'nas't͡s']
сакратар	NOUN	[сакратар́]	[sakratár]
сакратарка	NOUN	[сакратар́ка]	[sakratárka]
сакратарства	NOUN	[сакратар́ства]	[sakratárstva]
сакратарыят	NOUN	[сакратарыйат́]	[sakrataryját]
сакрацін	NOUN	[сакрац'ін́]	[sakrat͡s'ín]
сакрэт	NOUN	[сакрэт́]	[sakrét]
сакрэтка	NOUN	[сакрэт́ка]	[sakrétka]
сакрэтнасць	NOUN	[сакрэт́нас'ц']	[sakrétnas't͡s']
сакрэтнік	NOUN	[сакрэт́н'ік]	[sakrétn'ik]
сакрэтнічанне	NOUN	[сакрэт́н'ічан':э]	[sakrétn'it͡ʂan':e]
сакрэцыя	NOUN	[сакрэц́ыйа]	[sakrét͡syja]
сакс	NOUN	[саќс]	[sáks]

Fig. 2. An excerpt of the phonetic dictionary in the NooJ format

Since every dictionary is a finite set of words and their descriptions, for phonetization of words not contained in the dictionary, morphological NooJ grammars were developed. These grammars represent rules for the multivoice TTS system phonetic processor in the form of a finite automaton (Fig. 3).

In Belarusian, one letter may be represented by different phonemes, depending on their surrounding letters or position in the word. The most common sound changes in Belarusian are assimilation, elision and positional fortition. For instance, in the word дуб 'dub – Eng. oak', the last letter B changes into the sound [p] as a result of end-word fortition. These sound changes are presented in the grammar as follows: all the graphemes, which are surrounded by other particular graphemes, are given as an output allophone match, for instance, grapheme B from the example above will be marked by P as a corresponding allophone.

4 Phoneme-to-Grapheme Conversion for the Belarusian NooJ Module

Speech recognition systems are complex technical systems in which information passes through a large number of processing steps and transformations. A typical system consists of two main parts: signal processing and natural language processing. Most of the recent publications are focused on questions related to signal processing, including such techniques as dynamic time warping, hidden Markovs models and neural networks [11, 12].

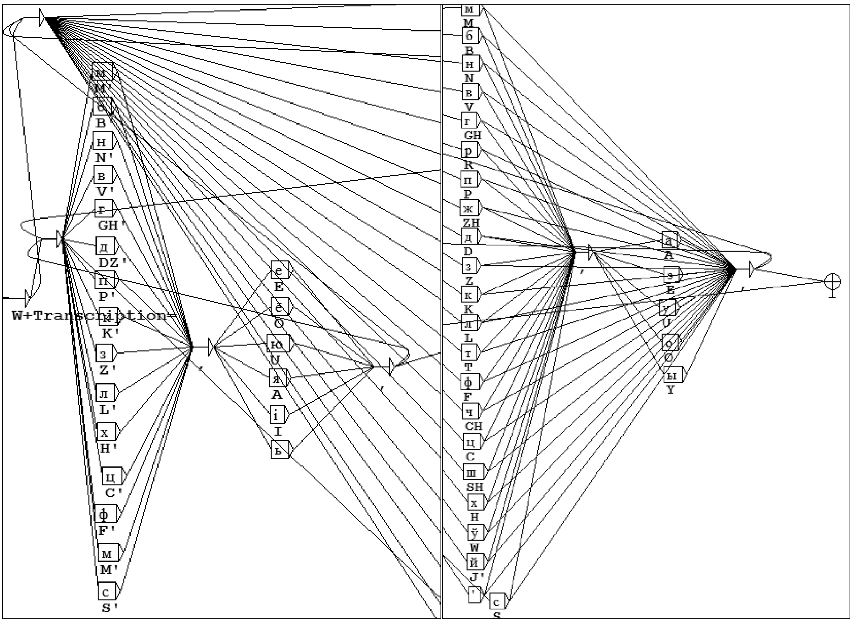


Fig. 3. An excerpt of the morphological NooJ grammar illustrating consonant softening (left) and hard consonants (right)

However, in the latest research, NLP is almost never mentioned. This observation is especially true for lesser known languages such as Belarusian. However, this part of the system contains many compelling challenges for researchers, which will provide ample opportunity for further research.

One of the basic tasks for developing the NLP part of a SST system is to transform phonetic elements into their graphemic representation (P2G). After acoustical decoding of separate phonetic units from feature vectors of a speech signal, phonemic text is obtained. This text is a source for the next language processing steps on morphological, syntactic and semantic levels. Therefore, effective implementation of phoneme-to-grapheme transformation is the first important task in the construction of the NLP part of a STT system. We would like to show how this problem could be effectively solved with the help of NooJ instruments for the Belarusian language.

During the process of conversion P2G, all of the abovementioned language phenomena should be taken into account. Considering the peculiarities of the Belarusian language we made the following assumption: it is possible to build a compact system of rules, which would constitute the core of the conversion algorithm according to these rules, then extend it according to accounting-specific conversion rules and a dictionary for words not included in the basic set, as well as phonetic contexts.

Therefore, the above situations require further description of certain rules. We propose to use NooJ to develop grammars because they are extremely simple objects to build, and there is no complex formalism to learn. NooJ includes tools to check, debug, adapt, maintain, and share language resources. NooJ can effectively solve the problem with the development of rules (linguists' responsibility) and then by conversion into a useful form for further formalization in terms of algorithm and implementation in the code (programmers' responsibility). This is especially important for non-engineering staff (linguists, phoneticians), as NooJ provides a user-friendly graphical interface which is easy to learn and use, in contrast, for example, with a regular-expression language.

We offer a solution to the problem of converting P2G using NooJ in two stages. The first stage is based on the rules for conversion using a NooJ grammar. The grammar contains the set of rules for the phoneme-to-grapheme conversion of the following classes:

- Rules for the base cases of phoneme-to-grapheme mapping. They use the principle of a return to the basic rule based on the phonetic spelling principle. These G2P rules can be relatively easily transformed into the rules designed to reverse conversion. For example:
  - [*d͡ʐ*] – дж;
  - [*γ*] – г.
- The rules for considering cases of combinatorial changes (assimilation, accommodation, dissimilation sounds) the interaction between the phonemes of the language and their rules. For example:
  - [*'sʲvʲata*] – свята (holiday);
  - [*'d͡zʲɛt͡sʲi*] – дзеці (children).
- Rules take into account positional changes as well as the effects (diarezis, epintezis, konraktasis) and phenomena occurring at the junction of phonetic words. For example:
  - [*'va'wlad͡zʲɛ*] – ва ўладзе (in power);
  - [*ɣara't͡skʲi*] – гарадскі (city *adv.*).

The second stage of the conversion is carried out using a dictionary approach to define exception words and resolve conflicts on the morphological and syntactic levels. We should also use an exception lexicon and even go beyond words to consider the context. The exception lexicon contains the pronunciations for irregular words. If we find such a word in the exception lexicon, we get the correct spelling at once and the two previous steps can be skipped. One more and very important task involves homonyms (words that share the same pronunciation but have different meanings). Consider the following case:

- [*pra'ɣramnɨ*] [*'kɔt*] – праграмны код (programming code);
- [*pu'ʂɨstɨ*] [*'kɔt*] – пушысты кот (fluffy cat).

In this example, the words ***код*** (*code*) and ***кот*** (*cat*) are phonetically identical. In this case, the G2P conversion of words is only possible by analyzing beyond the boundaries of the phonetic elements of speech. Namely,this can be done by semantic analysis of these fragments.

Currently, we are developing the first rule-based stage of the proposed approach. We assume that a real STT system P2G conversion process, executed by phonetic-decoding block (Fig. 1), will include the following basic steps (Fig. 4):

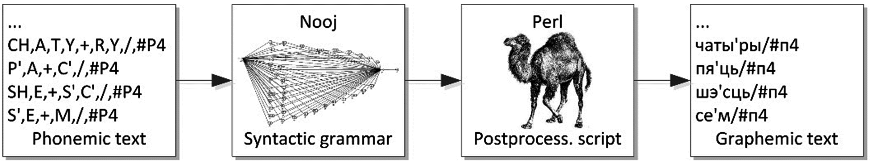


Fig. 4. Scheme of the phonetic decoding process in the context of STT

From the acoustic decoding block of STT, we obtain preliminary information in the form of an allophone sequence. Allophones are positional and combinatorial variants of phonemes and can also be used for P2G conversion. However, the sequence of allophones is not suitable for conversion into orthographic words directly, because of the large number of possible combinations. Thus, it is necessary to reduce the spatial dimension of this sequence. To do this instead of handling the direct allophonic sequence, we convert it into phonemic text. This problem can also be solved with the help of NooJ. However, implementation of this task is not within the scope of this publication, so the details are not presented here. This raw allophonic sequence from the acoustic decoding block is preprocessed to get a phonemic text. This text is converted into one of the text formats supported by NooJ and is then loaded into the system for the further processing.

Firstly, the algorithm detects phrases and sentences, replaces pauses, collapses sentences to phonetic words and finally generates the list of phonetic elements. Then the system processes each element in this list to get grammatical annotations using NooJ syntactic grammars. The main graph of one of the core grammars is presented in Fig. 5.

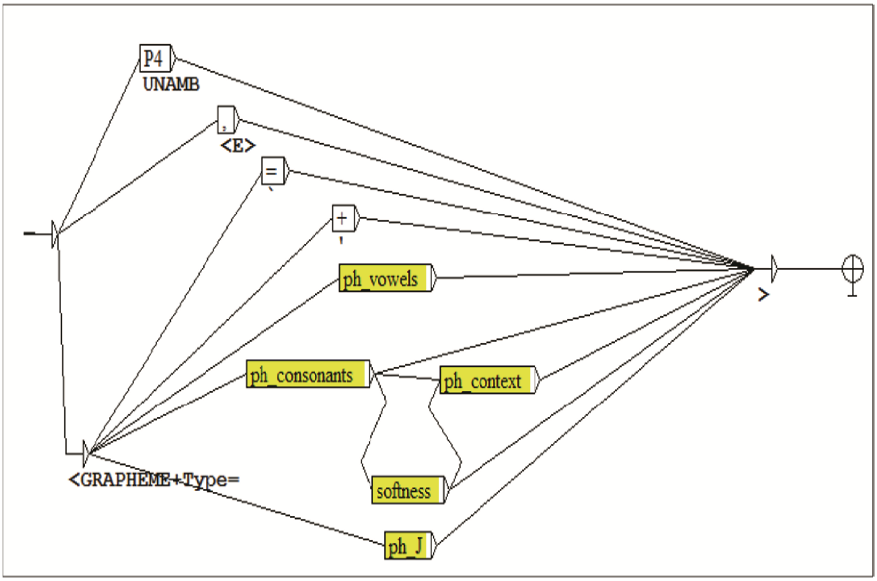


Fig. 5. Grammar of the upper level

One of the features of our work is that we used a grammar of a syntactic not a morphological level for the processing of phonetic units. This is due to the fact that the acoustic decoding block can generate the phonetic sequence in different formats (different types of separators, etc.). A syntactic level grammar allows us to consider phonemes as stand-alone words in a given environment and also allows for more flexibility. The grammar on the top level includes a set of sub-graphs for the left and right context for the given symbol and allows us to consider the cases described above, which can be executed recursively. One of the sub-graphs is presented in Fig. 6.

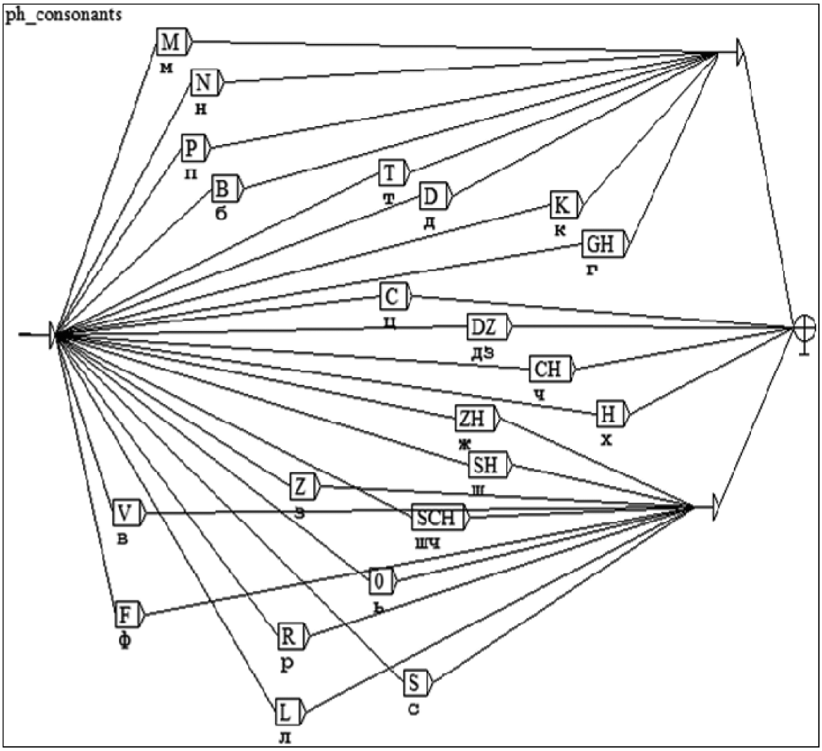


Fig. 6. One of the sub-graphs representing consonant conversion rules

The result of the conversion is exported to an XML file. Then it is processed with a special Perl script, which transforms NooJ annotations into general orthographic text, makes a plain text and saves it as an HTML file.

An example of grammatical annotation to be converted into graphemic form:

```
<GRAPHEME Type="л">L</GRAPHEME><GRAPHEME Type="">,</GRAPH
EME><GRAPHEME Type="і">I</GRAPHEME><GRAPHEME Type="">,</G
RAPHEME><GRAPHEME Type="'">+</GRAPHEME><GRAPHEME Type="">
,</GRAPHEME><GRAPHEME Type="к">K</GRAPHEME><GRAPHEME Type
="">,</GRAPHEME><GRAPHEME Type="і">I</GRAPHEME><GRAPHEME
Type="">,</GRAPHEME><GRAPHEME Type="">,</GRAPHEME><GRAPHE
ME Type="п">P</GRAPHEME><GRAPHEME Type="п">P</GRAPHEME>…
```

The suggested grammars are quite simple in form, but they allow us to quickly and easily perform the P2G conversion. Even if all the words are not fully recognized, this could be done later using a dictionary of exclusion words, which we plan to develop in the future.

5 Experimental Results

To determine the effectiveness of the linguistic resources developed for the Belarusian module of NooJ—one vocabulary and two grammars—a series of tests was performed. One test method involved the comparison of converted words with words from the reference source. If the resulting form did not coincide with the sample and had at least one error, it was marked as incorrect but converted. The errors committed by the algorithm were calculated for each test word separately and for the entire sample as a whole. In order to evaluate the effectiveness of the algorithms and the proposed grammars, we used the metrics of precision and recall described in [13]. *Precision* (P) is equal to the number of cases where conversion was correct (M) divided by the total number of phonetic cases retrieved by grammars (L). It can be determined according to the expression:

$$P = M/L.$$

Recall (R) is equal to the number of cases where conversion was correct (M) divided by the number of valid cases predetermined by expert assessment (N):

$$R = M/N.$$

It should be noted that these two quantities are trade off one against another. Recall is a non-decreasing function of the number of cases retrieved. On the other hand, in a good system, precision usually decreases as the number of cases retrieved is increased. In general, we want to get some amount of recall while tolerating only a certain percentage of false positives. To provide unbiased evaluation, it is desirable to use a certain metric that balances the previous two. *Weighted harmonic mean* (F) is a single measure of precision versus recall compromise degree:

$$F = 2 * P * R/(P + R),$$

where P and R are *precision* and *recall*, respectively. Recall, precision and F measure have a values between 0 and 1, but they are also very commonly written as percentages, on a scale between 0 and 100 %. The experimental results are presented in Table 4.

Table 4. Error rates of G2P and P2G conversion process experiments

Conversion type	Metric type	Part of speech					
		Noun		Verb		Adjective	
		I	II	I	II	I	II
G2P	P	0,90	0,88	0,91	0,90	0,92	0,89
	R	0,91	0,82	0,93	0,87	0,85	0,83
	F, %	90	86	92	88	87	85
	Avg. F, %	88		91		86	
P2G	P	0,87	0,86	0,84	0,82	0,90	0,86
	R	0,81	0,85	0,84	0,86	0,79	0,79
	F, %	84	83	84	84	84	82
	Avg. F, %	85		84		83	

To carry out a sample test, information from three sources was used, among them the Belarusian phonetic dictionary, the English-Belarusian phrasebook,“ Orthoepic Dictionary Generator”, and linguistic resources from “corpus.by” project [7, 14–16]. The size of the test corpus is 2400 words. It was formed in a special way for testing G2P and P2G conversion algorithms for Belarusian. It is representative in terms of lexical content and also phonetically balanced. The corpus includes both individual words and phrases, as well as whole sentences. Since the character is the basic unit of transformation and the average length of the corpus element is about 9,5 symbols, the total size of the test sample is about 22800 characters. The corpus was divided into three subsets of different size, depending on the part of speech: nouns (1000 words), verbs (800 words) and adjectives (600 words). Each subset in turn was divided into type I and type II groups, depending on the frequency of occurrence of words in the texts. The average F-measures for G2P and P2G conversion were grouped by part of speech as presented in the following histograms (Fig. 7).

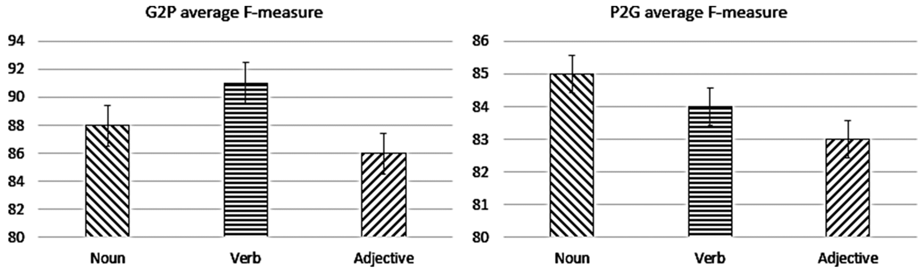


Fig. 7. Average error of G2P (left side) and P2G (right side) conversion

Analysis of the results leads to some compelling conclusions. Firstly, the F-measures of G2P conversion are on average 5 % higher than the measures for P2G conversion. This can be explained by the presence of well-defined rules for Belarusian for conversions of the first kind; however, the rules are not always suitable for conversion in the opposite direction. Secondly, there is some sort of correlation between error rates and part of speech. The average length of adjectives in the Belarusian language is more than for other parts of speech, so it is natural that more errors were observed, which means that more attention and highly complex transformation rules are needed for the handling of exceptions. The third largest component of the total conversion error is determined by the part not yet covered by the rules and not covered by the dictionary of exceptions. As such, it is the aim of our future research to reduce errors and improve the quality of conversion. On the other hand, the number of direct conversion mistakes is not so high, so we can conclude that the grammar we developed showed good results.

6 Conclusion

As part of this ongoing research, procedures and algorithms for the conversion of G2P and P2G processes using a linguistic development environment NooJ have been developed. A dictionary containing words and their transcriptions was made. A morphological transformation grammar and rules for processing written words in phonetic form and for the reverse transformation were built.

The results of this research could be used to solve various computer-linguistic problems: forward and reverse phonetization, adding phonetic level to information processing in the Belarusian NooJ module, and developing a SST system. In addition, we are planning to use the G2P dictionary and grammar in the multivoice TTS system for the Belarusian language [17].

It should be noted that one of the main objectives of this work was also the author's desire to show the capability of NooJ tools to work on various linguistic levels including the phonetic representation of written language. NooJ tools can be fully utilized for processing all kinds of texts, including phonetic representation.

Future work will involve adding to and expanding the existing system of rules for G2P conversion as well as the addition of exceptions and special cases for the P2G conversion process and further integration of these results into the Belarusian module of NooJ.

References

1. Silberztein, M.: NooJ Manual (2003). www.nooj4nlp.net
2. Marchant, C.C.: Fundamentals of Modern Belarusian (2004). http://www.vitba.org/fofmb/introduction.html
3. Сцяцко, П.: Уводзіны у мовазнаўства. Вышэйшая школа, Мінск (2001)
4. Андарала, Г.: Сучасная беларуская мова: фанетыка. БГУ, Мінск (2013)
5. Transcription Generator. http://corpus.by/transcriptionGenerator/. Accessed 6 Oct 2015
6. Падлужны, А.І.: Фанетыка беларускай літаратурнай мовы. Навука і тэхніка, Мінск (1989)
7. Кошчанка, У.А.: Беларуска-англійскі размоўнік. Артыя Груп, Мінск (2010)
8. The International Phonetic Alphabet and the IPA Chart. https://www.internationalphoneticassociation.org/content/ipa-chart. Accessed 6 Oct 2015
9. Computer-coding the IPA: a proposed extension of SAMPA. http://www.phon.ucl.ac.uk/home/sampa/x-sampa.htm. Accessed 6 Oct 2015

10. Text-to-Speech PHP-Based Synthesizer. http://corpus.by/tts3. Accessed 6 Oct 2015
11. Dutoit, T., Marques, F.: Applied Signal Processing: A Matlab-based Proof of Concept. Springer Science, Buisness Media, LLC, New York (2009)
12. Mitkov, R.: The Oxford Handbook of Computational Linguistics. Oxford University Press, Oxford (2005)
13. Manning, D., Raghavan, P., Shutze, H.: An Introduction to Information Retrieval. Cambridge University Press, Cambridge (2009)
14. Слоўнік граматычна-лінгвістычнае тэрміналёгіі (проект). Менск: Інбелкульт (1927)
15. Пашкевіч, В.: Ангельска-беларускі слоўнік/English-Belarusian Dictionary. Менск (2006)
16. Orthoepic Dictionary Generator. http://corpus.by/transcriptionGenerator/. Accessed 6 Oct 2015
17. Zahariev, V., Petrovsky, A., Lobanov, B.: Multivoice text to speech synthesis system. In: 12th International Conference on Pattern Recognition and Information Processing (PRIP 2014), Conference Proceedings, 28–30 May 2014. UIIP NASB, Minsk, pp. 320–324 (2014)