INFORMATICA, 2006, Vol. 17, No. 4, 587–600

Automatic Transcription of Lithuanian Text Using Dictionary

Mantas SKRIPKAUSKAS, Laimutis TELKSNYS
Institute of Mathematics and Informatics
Akademijos 4, 08663 Vilnius, Lithuania
e-mail: mantas.skripkauskas@ktk.lt, telksnys@ktl.mii.lt

Received: January 2006

Abstract. There is presented a technique of transcribing Lithuanian text into phonemes for speech recognition. Text-phoneme transformation has been made by formal rules and the dictionary. Formal rules were designed to set the relationship between segments of the text and units of formalized speech sounds – phonemes, dictionary – to correct transcription and specify stress mark and position. Proposed the automatic transcription technique was tested by comparing its results with manually obtained ones. The experiment has shown that less than 6% of transcribed words have not matched.
Key words: speech recognition, grapheme to phoneme transcription.

1. Introduction

Both tasks of speech recognition and synthesis are related to some transformation of the textual information into phonemes. For example, the speech recognition is nothing else but writing acoustical speech signal in the text, or speech synthesis – conversion of the text into acoustical signals. To make it possible there must be known what sounds relate to correspondent parts of the text, i.e., there is a need to have transcriptions of all analyzed words. There are over 50 000 words collected in "The Dictionary of Modern Lithuanian Language" (Keinys, 2000), and over 11 million headwords in twenty toms of (Dictionary of the Lithuanian Language (vol. I–XX, 1941–2002): electronic version, 2005). Because words of The Lithuanian language can have a lot of different variable forms (e.g., nouns, adjectives, numerals, pronouns can vary because of cases, singular or plural forms, or other factors, and verbal words vary because of different tenses, conjugations, moods or other reasons (Ambrazas, 1994)), then overall number of words will be much bigger. If this task would be needed to be done manually, then we would be having very labour-intensive work, because it is known from practice, that a one work day is needed to transcribe one thousand words, when not only phonemes, but also the accents, and consonant palatalizations are marked. When the whole process is being done by one man it would take 25 years to transcribe the headwords contained in mentioned dictionaries and to transcribe all variable forms of those words would take several thousand years. Because of this, the transcription process needs to be automated. The big-

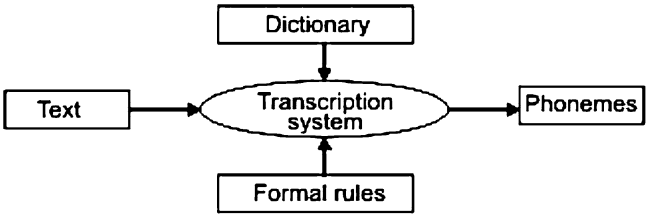

Fig. 1. Automatic text transcription system.

ger part of those words can be easily transcribed using the Lithuanian language pronunciation rules (Ambrazas, 1994), but they have a lot of exceptions. For example, foreign words (e.g., "liana", "liapsusas", "liatas"), which in the computer-aided dictionary (Vaitkevičiūtė, 2001b) are over 20 000 most popular words, in Lithuanian language terminology dictionary ("Terminology of Lithuanian Language", 2005) are over quarter of a million headwords. Next would be various abbreviations or acronyms (Marcinkevičienė, 2003) (e.g., "D6", "AIDS", "NATO", "NKVD"), compound words (e.g., "abiakis", "angliadegys"), and proper words (e.g., "Ariadnė", "Izaijas"). Hence, we can not rely only on the pronunciation or so called formal pronunciation rules (Kasparaitis, 1999), therefore we will need the additional pronunciation dictionary in order to transcribe correctly exceptional words.

Systems (Fig. 1), which transcribe text into the phonemes, when formal pronunciation rules and the dictionary are in use, have been already created for other languages. But they can not be used in transcribing Lithuanian text, because of natural differences of languages and of used different phonetic units. Examples of such systems are presented in proceedings like (Torstensson, 2002), in which various problems of the transcription process and their solutions are described. Main problems of the transcription process were: formation of the formal pronunciation rules (Torstensson, 2002), mapping analyzing word with the dictionary headword (Stoianov, 2001), correcting the word transcription using acoustical signal (Decadt, 2001), foreign word pronunciation (Stoianov, 2001), transcription formation using statistical methods (HMM, finite state transducers (Uneson, 2005, Caseiro, 2002), ANN (Stoianov, 2001)) or decision trees (ID3 (Demuynck, 2002), CART (Quazza, 2001), C.45 (Baldwin, 2000) and other). Details of all those methods can be found in (Stoianov, 2001).

The transcription of the Lithuanian text has been studied in (Kasparaitis, 1999), where were presented few of the many possible principles to form formal pronunciation rules for transcription of the Lithuanian text. This work was more oriented into synthesis of the Lithuanian speech, because of characteristics of the used phonetic unit. For example, phonemes of the consonants were modeled not like separate phonemes, but like combinations with silence or other sounds of speech (e.g., "ka", "ke" and other). Also the experiments of the designed automatic transcription system have not been done.

Hence, in this paper we propose the different transcription system of the Lithuanian text, which was firstly intended for the Lithuanian speech recognition, and secondly, it was based on the combination of formal pronunciation rules and the dictionary. Also

there will be given results of the experiments, which were performed with the system, and conclusions were obtained. Main differences from the system presented in (Kasparaitis, 1999) were usage of phonetic units, formation of the formal pronunciation rules and the dictionary. Also this system has been tested and results of the experiments were provided. Formal pronunciation rules, as in other language systems, were created by using various rules, which were made by Lithuanian linguists (Ambrazas, 1994, Vaitkevičiūtė, 2001a). And the dictionary contains accent and a lot of word pronunciation characteristics, which are as exceptions to the formal pronunciation rules. It was made of other three dictionaries: "The Dictionary of Modern Lithuanian Language" (Keinys, 2000), "Terminology of Lithuanian language" ("Terminology of Lithuanian Language", 2005) and V. Vaitkevičiūtė "Dictionary of foreign words" (Vaitkevičiūtė, 2001b).

2. Theoretical Part

As it was mentioned earlier, the system, which is presented in this work, was composed from two parts: formal pronunciation rules and the dictionary. Formal rules were used to describe Lithuanian pronunciation rules, the dictionary – to identify the exceptions of these rules and to get additional information, e.g., stress. The inclusion of the dictionary to the system have let better transcribe the wider format text, for example, broadcast news, scientific reports, which contain a lot of:

1. foreign words (e.g., "liana", "liapsusas", "liatas"),
2. acronyms (e.g., "D6", "AIDS", "NATO", "NKVD"),
3. compound words (e.g., "abiakis", "angliadegys"),
4. Proper nouns (e.g., "Ariadnė", "Izaijas").

2.1. *Formal Pronunciation Rules*

Formal pronunciation rules or just formal rules describe relationship between phonemes and segments of the text (e.g., word "lankas" is converted by pronunciation rules into the phoneme group "l aw k a s"). In every language this relationship is different. In the Lithuanian language usually all letters in the text are pronounced, but there are exceptions, which are described and solved with the help of formal rules. Formal rules are made of pronunciation rules of an analyzed language, which can be found in various grammar books (Ambrazas, 1994, Vaitkevičiūtė, 2001a). An example of the formal rule could be a union of letters "ar", which in the text will be phoneme "ar", when after letter "r" there will not be any vowel. In other cases this union will match two phonemes "a" and "r".

As it was mentioned earlier the Lithuanian text can be read by letters with some exceptions, which are described by formal rules, the transcription can be made by simply modifying the written text (e.g., transcription for word "laksto" will be "l a k s t o"). So we see that all phonemes in the transcription are separated by space mark and their names are one or more spoken letters, which usually can be found in that text. There are five text modification operations, from which the transcription is obtained:

1. separation, when letters or their combinations in the text, which labels phonemes, are separated from other adjacent letters-phonemes (e.g., word “palis” is resolved in separated phonemes “p”, “a”, “l”, “i”, “s”),
2. concatenation, when several letters are concatenated into one phoneme (e.g., two adjacent letters “a” and “u” in word “kaupti” are concatenated into one phoneme “au”, then full phonetic set of this word will be “k”, “au”, “p”, “t”, “i”),
3. deletion, not pronounced letters are removed from the transcription (e.g., one letter “š” in the word “iššovė” is removed, and all others form separate phonemes “i”, “š”, “o”, “v”, “ė”),
4. insertion, when additional letters are inserted into the transcription, which labels pronounced phonemes, i.e., not written by any letter, but existing for some reason (e.g., additional phoneme “j” appears in word “ietis”, so the full set of phonemes will be “j”, “ie”, “t”, “i”, “s”),
5. and replacement, when there is such a combination of letters in the word, pronounced differently from written combination (e.g., combination of letters “gt” in word “bėgti” needs to be replaced by “kt” and respectively full set of phonemes of the word will be “b”, “ė”, “k”, “t”, “i”).

Every operation mentioned above is accomplished strictly by the formal rules, which can be described this way:

1. lets mark a set of text symbols $A = \{a_1, a_2, \ldots, a_N\}$ as seeking symbols or their combinations,
2. analyzed text word mark as $W = \{w_1, w_2, \ldots, w_M\}$, where $w_1..w_M$ are word letters,
3. mark replacement symbols as $X_i = \{x_1, x_2, \ldots, x_N\}$,
4. $B = \{b_1, b_2, \ldots, b_M\}$ is a special set of symbols, which shows when do we need to change seeking symbols a_n by respective X_n symbols,
5. thus, the full rule can be written this way:
   a) if we have a combination of symbols $\{w_{t-r}, \ldots, w_{t-1}, w_t, w_{t+1}, \ldots, w_{t+f}, [b_j]\}$ (here units in angle brackets are optional, and not in brackets – required), in which the main examined symbol is w_t, and $w_{t-r}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+f}$ – context symbols,
   b) in which $a_i = \{[w_{t-r}, \ldots, w_{t-1}]w_t[w_{t+1}, \ldots, w_{t+f}]\})$, then a_i is replaced by X_i.

Then text modification operations can be written in these two finite state transducers (FST) (Roche, 1997):

1. Separation operation
   P= $\{\Sigma$, Q, i, F, E$\}$, where P – finite state transducer, Σ – phonetic unit set, Q= $\{$fs$_0$, s$_1\}$ – state set, $i = fs_0$ – initial set, $F = \{fs_0\}$ – finite state set and E – state transition set.

$$E = \{ \quad fs_0 : \{\Sigma\} : s_1 \\ fs_0 : \text{“ “} : s_1, \\ s_1 : (\varepsilon : \text{“ “}) : fs_0 \\ \}.$$

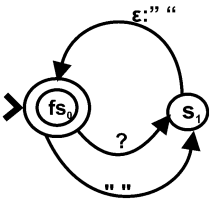

Fig. 2. FST of separation operation.

$[\{\Sigma\} \rightarrow \varepsilon \ldots \text{“ “}]$ – regular expression of the given finite state transducer. In this expression $\{\Sigma\}$ and in graph "?" means any phonetic unit, ε – empty set, "$\rightarrow \varepsilon \ldots$ " " " marks operation, that space symbol needs to be put after any phonetic unit.

2. Deletion, insertion and replacement operations
   The first two operations can be described as replacement operation. That is, for deletion operation the searched symbol has to be replaced by empty set, and for insertion – by concatenation of the searched and inserted symbols.
   Replacement operation can be written in formal way by this finite state transducer: $P = \{\Sigma,\ Q,\ i,\ F,\ E\}$, where again P – finite state transducer, Σ – phonetic unit set, $Q = \{fs_0, fs_1, s_2\}$ – state set, $i = fs_0$ – initial set, $F = \{fs_0, fs_1\}$ – finite state set and E – state transition set.

$$E = \{ \quad fs_0 : \{\Sigma\} : fs_0, \\ fs_0 : a_i : fs_1, \\ fs_0 : X_i : fs_0, \\ fs_0 : b_j : fs_0, \\ fs_0 : (a_i : X_i) : s_2, \\ fs_1 : \{\Sigma\} : fs_0, \\ fs_1 : a_i : fs_1, \\ fs_1 : X_i : fs_0, \\ fs_1 : (a_i : X_i) : s_2, \\ s_2 : b_j : fs_0 \\ \}.$$

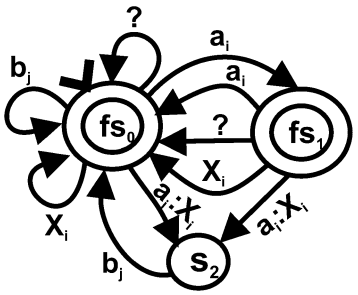

Fig. 3. FST of replacement operation.

$[a_i \rightarrow X_i || \varepsilon \_ b_j]$ – regular expression of the given finite state transducer. Operation "$\rightarrow X_i || \varepsilon \_ b_j$" in this expression means, that the unit a_i needs to be changed by X_i, if the unit b_i goes right after a_i. FST in Fig. 3 has three states. Two (marked with double circle) of them are finals. The arcs from one state to another means possible transitions between states. At each transition we have a pair of input and output parameters, this is noted as $a_i : X_i$, where a_i – input and X_i – output parameters. If we observe same input and output in one transition, than we just write one of them near the arc (e.g., $b_j : b_j \rightarrow b_j$). The starting state is labeled by arrowhead near the state, in this case it is fs_0 state. Now let us assume that we have a sequence of symbols "abbc" and we need to replace symbol "b" by "e", if after "b" is symbol "c". In this case parameter a_i is equal to "b", X_i – "e" and b_j – "c". Then the sequence "abbc" will be modeled by FST by these steps:

a) first symbol of the sequence "a" is not defined in replacement rule, so it will be denoted as "?" or $\{\Sigma\}$ and first transition from state fs_0 to fs_0 will be made $(fs0 : \{\Sigma\} : fs0)$,
b) second symbol "b" is equal to a_i and it can be or not replaced by X_i, therefore there are two transitions from state fs_0. One transition to $fs_1(fs_0 : a_i : fs_1)$ – meaning that we will not change it, and second to $s2(fs_0 : (a_i : X_i) : s_2)$ – meaning we will change it. In our case we will go to state fs_1, because third symbol is not "c",
c) in third step we will go to state s_2 by replacing third symbol "b" by "e" $(fs_1 : (a_i : X_i) : s_2)$,
d) and in final step we will return to our start state by observing the parameter b_j, which is equal to "c" $(s_2 : b_j : fs_0)$.

If we would like to describe the situation, when the unit b_j goes before a_i, then in the regular expression ε and b_i must be counterchanged. FST of this operation will have a bit more changes. Firstly, there will be only two states and both of them final. Secondly, there will be only seven transitions: $fs_0 : \{\Sigma\} : fs_0, fs_0 : a_i : fs_0, fs_0 : X_i : fs_0, fs_0 : b_i : fs_1, fs_1 : (a_i : X_i) : fs_0, fs_1 : \{\Sigma\} : fs_0$ and $fs_1 : a_i : fs_0$. Here we have presented the example of substitution. The other operations are processed in the same way, the difference of the operations lies in definition of parameter X_i. If we want remove letter, than parameter X_i will be empty set, and if – insert, than X_i will be equal to a_i plus letters needed to insert. More details of FST shown in Fig. 3 and similar to it can be found in (Karttunen, 1997).

But formal rules not always can help us. In particular they do not help, when we run into non-Lithuanian proper nouns or acronyms. So, to lower errors made by formal rules, when dealing with such words, we could use in advance made dictionary with pronunciations of words.

2.2. *Dictionary*

The main purpose of the dictionary is not only to correct the pronunciation of the problematic words, but to give the information about pronunciation features of transcribed words, e.g., accent. The biggest problem of using such a dictionary is to apply obtained information of headwords, which are in the dictionary, to the transcribed words. This is especially difficult in situations, when analyzed words can have a lot of variable forms. Because of this, there is a need to choose some ways, how to relate the analyzed word and the headword. This problem can be solved in several ways:

1. hold all forms of headwords in the dictionary,
2. hold only the main form of the headword, for example if the word is noun, adjective or numeral, then it is in singular nominative, if the word is a verb, then it is in infinitive form; and to have a mechanism, which can transform analyzed word into its main form,
3. hold such headwords in the dictionary, that can have any of its form, then the analyzed word can match that headword, which has the biggest number of matching letters between those two words starting from the beginning of words,

4. have the same dictionary as in the third case, but a relationship between analyzed word and headword is made with the help of various string search methods (Tillmann, 2000, Bentley, 1997) .

3. Practical Part

The set task of this work was done practically by creating the automatic transcription system of the Lithuanian text on the ground of two parts: formal pronunciation rules and the dictionary. When the system was designed, it was more oriented into the journalistic common Lithuanian speech, because of those formal rules, the dictionary, and main experiments were done with the fragments of that kind of language.

3.1. *Used Formal Pronunciation Rules*

Main formal pronunciation rules, which have been used in this transcription system, were formed by the work of V. Vaitkevičiūtė "Pronunciation background and dictionary of Lithuanian Language" (Vaitkevičiūtė, 2001a), because there they are written straight in orderly fashion. There were collected 400 formal rules at all. The most popular ones were related to subjects like:

1. Assimilation, when two consonants, voiced and unvoiced or vice versa, collides, then the first one from the left is changed by appropriate unvoiced or accordingly voiced one (e.g., "apdaila" → "a b d ai l a").
2. Diphthongs, when two appropriate vowels go together (e.g., "a" and "u"), then they are written like one phoneme in transcription (e.g., "laukas" → "l au k a s"),
3. Vowel and consonant combination, when we have a combination of three letters, where the first is one of the vowels "a", "e", "i", "u", the second symbol – one of the consonants "l", "m", "n", "r", and third one is some consonants, then the first two letters are written like one phoneme (e.g., "randas" → "r an d a s", "garas" → "g a r a s").
4. Palatalization vowels, when we have a combination of symbols, where the first ones are consonants, and the last symbol is one of the palatalization vowels "i", "į", "y", "e", "ę", "ė", then all those consonants are labeled with the apostrophe mark, which means, that a consonant is pronounced softly (e.g., "liepti" → "l' ie p' t' i").
5. Softness mark, letter "i" sometimes means not a vowel, but softness mark, which means, that all consonants, that go right before it, are pronounced softly, and the letter "i" itself is not pronounced (e.g., "kiauras" → "k' au r a s").
6. Long vowels, that are vowels "ą", "ę", "į", "y", "ų", "ū", "ė" and all stressed with acute and circumflex vowels are changed accordingly into phonemes "a:", "e:", "i:", "u:", "o:" and "ė:" (e.g., "rūko" → "r u: k o").
7. Adjacent consonants, when two same consonants go one beside another, then they are replaced by one phoneme made of the second letter (e.g., "iššoko" → "i š o

k o"); when any combination of two fricatives made of "s", "š", "z", "ž" go one beside another, than the first fricative is skipped and the other one is left (e.g., "išstumdavo" → "i s t um d a v o"); when there are two adjacent fricatives, where the first one is "ž" or "s" and the second – "č", than the first fricative is changed to "š", similarly if the first fricative is "z" and the second one – "dž", than the first one is changed to "ž" (e.g., "apsčiai" → "a p' š' č' ei", "vabzdžiai" → "v a b' ž' dž' ei").

8. "n" before "k" or "g", when we have a combination of two letters, where the first one is letter "n" and the second – "k" or "g", then the letter "n" is pronounced as a nasal sound and marked as the phoneme "w" (e.g., "anga" → "aw g a").
9. Transformation of single or group of capital letters into phonemes, when we have a group of letters, mostly capitals or words contained numbers, which needed to be pronounced as single letters, not as combinations (e.g., "NKVD" → "en k a v' ė d' ė").
10. Transformation of numbers into phonemes, when numbers in the text are changed according words (in singular nominative) and transcribed (e.g., "D-6" → "d' ė š' e š' i").
11. Other rules.

3.2. *Used Phonemes*

The final result of this work was a collection of phonemes, which matched some analyzed text. Because the text transcription was made by using V. Vaitkevičiūtė's pronunciation rules, phonetic units were written the way the authors did in her work (Vaitkevičiūtė, 2001a), only labels of stress and some letters were modified.

Hence, using formal pronunciation rules the system converts the text into 225 phonemes, which can be classified in these two groups:

1. Phonemes made of one pronounced sound (68 phonemes):
   a) vowels: " a a0 a: a:1 a:2 e e0 e: e:1 e:2 e3: e3:1 e3:2 i i0 i: i:1 i:2 o o0 o:1 o:2 u u0 u: u:1 u:2 ",
   b) unvoiced consonants: " p p' t t' k k' s s' s2 s2' f f' ",
   c) voiced consonants: " b b' d d' g g' z z' z2 z2' " ,
   d) other consonants: " c c' c2 c2' h h' j' l l' m m' n n' w w' r r' v v' ".
2. Phonemes made of two inter-merged sounds:
   a) diphthongs (22 phonemes): " ai ai:1 a:2i au au:1 a:2u eu eu:1 e:2u ui u0i ui:1 u:2i uo uo:1 u:2o ie ie:1 i:2e ei ei:1 e:2i ",
   b) mixed diphthongs (130 phonemes): " al al1 al' al'1 a:2l a:2l' am am1 am' am'1 a:2m a:2m' an an1 an' an'1 a:2n a:2n' aw aw1 aw' aw'1 a:2w a:2w' ar ar1 ar' ar'1 a:2r a:2r' el e0l el1 el' el'1 e0l' e:2l e:2l' em e0m em1 em' e0m' em'1 e:2m e:2m' en e0n en1 en' e0n' en'1 e:2n e:2n' ew e0w ew1 ew' e0w' ew'1 e:2w e:2w' er e0r er1 er' e0r' er'1 e:2r e:2r' il i0l il1 il' i0l' il'1 im i0m im1 im' i0m' im'1 in i0n in1 in' i0n' in'1 iw i0w iw1 iw' i0w' iw'1 ir i0r ir1 ir' i0r' ir'1 ul u0l ul' ul1 u0l' ul'1 um u0m um1 um' u0m' um'1 un u0n un1 un' u0n' un'1 uw u0w uw1 uw' u0w' uw'1 ur u0r ur1 ur' u0r' ur'1",

c) other diphthongs (5 phonemes): " dz dz' dz2 dz2' " (are assigned to the group of voiced consonants) and " ch ".

Some control symbols were used to write these phonemes, also Lithuanian letters were encoded using a combination of Latin letters and numbers. Phonemes, which were encoded by sequence of symbols "c2" meant "č", "e3" – "ė", "s2" – "š", "z2" – "ž", "dz2" – "dž". Symbol "w" in phonemes labels such phonemes "n" situation, when it goes before one of "k" or "g" phonemes. The number in phoneme shows that it is stressed, except the case, when we have Lithuanian letters. The number "0" means grave, "1" – circumflex and "2" – acute stress mark. The stress mark in a phoneme always goes after the stressed phoneme letter and its other differential features. If the phoneme is made of one sound, then stress mark will always be last, if of two sounds, then the stress mark will be last, when stressed sound is the second, or right before letter, which labels second sound, when the first sound is stressed. The colon mark, as was mentioned earlier, is put only near vowels, and means that this vowel needs to be pronounced long, because of its nature, or it was stressed with circumflex or acute stress. Apostrophe mark shows that a symbol must be pronounced softly and is put only near consonant letters.

3.3. *Used Dictionary*

As it has been already mentioned, the dictionary was used to make the word transcription. With the help of the dictionary, it was possible not only to correct the transcription, but also to get additional information, e.g., how the analyzed word is stressed.

The every row of the dictionary consists of two words. The first one was used as a headword, and the second one – as a pronunciation correction of the headword, where we can also find a stress mark and position. Thus, the dictionary was made of more than 220 000 rows.

Words in this dictionary were collected from three sources:

1. The Dictionary of Modern Lithuanian Language (50 000 words) (Keinys, 2000)
2. Terminology of Lithuanian language (250 000 words, but contains a lot of same words as in the first source) ("Terminology of Lithuanian Language", 2005)
3. V. Vaitkevičiūtė "Dictionary of foreign words" (20 000 words) (Vaitkevičiūtė, 2001b)

All three dictionaries were put into the one big dictionary, where repeated rows were removed from it. All words in the dictionary were made of combination of Latin letters and numbers, and in the second column of the dictionary apart from mentioned symbols there are five more additional symbols. Three symbols for stress ("/", "\" and "∼") and two symbols for labeling that two adjacent letters are independent ("!" means syllable boundary, and "@" – palatalization independence). The example of syllable boundary could be "trias2is tri!a∼s2is", where mark "!" is used to show that rule of the softness mark here cannot be implemented. The palatalization independence mark usage could be illustrated with "galva gal@va", where mark "@" means that consonant "l" in any variable form of the word is never soft (e.g. "galvyt" → "g al v' y t").

Headwords in dictionary do not have one single form, because words in Lithuanian can be pronounced very differently from the same word but in different variable forms

(e.g., “yra” – “buvo” or “eiti” – “ėjo”). Therefore it was tried to put a lot of various forms of words into the dictionary. Relation between transcribed word and dictionary headwords was made by the third way mentioned in Paragraph 2.2, where there relation between two words was made by the bigger number of the same symbols in relating words starting from the beginning of them. However, in order to avoid possible mistakes, thresholds were put two: a minimal number of matching letters and a number of letters that are in those words. So, if a number of matching letters is less than a half of a transcribed word or if numbers of letters in both words differ more than in a third of number of letters in transcribed words, then these words are not related.

Stress of the transcribing words was determined only by the dictionary. Because not all word forms were put into the dictionary, not all transcribed words have stress marks. It was decided to put stress marks only on those words, which can be found in the dictionary.

Seeking to reduce the number of mistakes, when dealing with prefixed or compound words, there was performed partial morphological analysis, in order to find words with prefix, reflexive part or more than one root. The set of 10 000 different Lithuanian roots and about 30 least prefixes were used for this procedure. All these data were taken from the composition analysis part of “The Dictionary of Modern Lithuanian Language” (Keinys, 2000).

4. Experiment

The created system was tested in transcribing four collections of words. In order to be sure, that the dictionary gives any gain, every collection was transcribed the twice the first time using the dictionary and the second time – without it.

The first collection was made of words taken from texts read by newsreaders of the “Lithuanian radio”. There were over 18 000 different words.

The second testing collection was made of 5000 different words which mostly appeared in nonfiction literature taken from “Text corpora of Lithuanian language” (Marcinkevičienė, 2004).

Words for the next testing collection were taken from the scientific paper in medicine field (Journal “Health science. Public health. Medicine. Nursing.”, 2005), in which there were a lot of various names of diseases, cures, healing methods and acronyms. There were more than 5200 various different words.

And words for the last collection were taken from informational system field of “Nonfiction literature” of “Text corpora of Lithuanian language” (Marcinkevičienė, 2004). Because system have let to get sets of 100 words by giving first letters of the wanted words, all testing collection consisted of 15 sets of that kind of words, which were intentionally selected by the authors and were not placed in the conclusions. There were more than 1500 different words. This collection was used only for making tested transcription system to do errors.

Transcription results of every collection are presented in Table 1, where we can see that transcribing with the dictionary occur almost twice less errors comparing with transcription without the dictionary in all four collections. The information about the number

Table 1

Results of experiments

		Percent of erroneously transcribed words	Number of matched phonemes	Number of erroneous phonemes	Total number of phonemes in automatic way	Total number of phonemes in manual way	Total number of different phonemes in automatic way	Total number of different phonemes in manual way
1 collection	With dictionary	0.9%	137771	677	138402	138448	237	236
	Without dictionary	2.2%	136424	2024	137935	138448	111	236
2 collection	With dictionary	0.8%	31207	206	31297	31413	229	230
	Without dictionary	1.4%	31042	371	31277	31413	110	230
3 collection	With dictionary	2.1%	41491	358	41733	41849	213	219
	Without dictionary	5.2%	40407	1442	41511	41849	107	219
4 collection	With dictionary	0.7%	12870	59	12915	12929	167	166
	Without dictionary	3.0%	12583	346	12847	12929	109	166

of phonemes gotten during the transcription persuades that these results were reliable. We can see that number of phonemes is twice less, than the one in transcription with the dictionary. This is normal because phonemes with stress marks are not included.

Main causes of errors were these:

1. transcription of unknown acronyms (LRT, NBA etc.),
2. transcription of proper nouns (various names, surnames etc.),
3. transcription of mistakenly written words (which come from writing down mistakenly pronounced speech) (e.g., “leiss2ta”).

Most of those problems can be solved by adding more proper noun and abbreviation words to the system dictionary.

By the time this publication was written, there were no published results of the other automatic Lithuanian text transcription system. But there was mentioned earlier that transcription system for The Lithuanian text-to-speech task was created by (Kasparaitis, 1999). The author of it agreed to do modifications, which were needed for comparison purposes, mostly, the descriptions of phonetic units and some formal rules had to be changed. There were three modification stages at all, i.e., there were three tries of the author to improve his system for the better results to be gotten. Results that show transcriptions systems quality, developed by P. Kasparaitis, are given in Table 2.

These results show that creation of the formal rules is not an easy task, though they are given in various linguists grammars. There are always some situation, which is not described by the formal rules, especially, when transcribed text is in the Lithuanian language. This can be seen by comparing results of the first and third stages. The results in the third stage have become better, only because the transcription situations, for which there was no formal rules or the rules where mistakenly written, have been found. The use of the dictionary in this task can considerably to lower the occurrence of those situations

Table 2

Kasparaitis transcription systems results

		Percent of erroneously transcribed words	Number of matched phonemes	Number of erroneous phonemes	Total number of phonemes	Total number of diffeent phonemes
1 stage	1 collection	14.9%	133285	5163	138437	249
	2 collection	12.9%	30291	1122	31429	217
	3 collection	22.6%	39661	2188	41776	208
3 stage	1 collection	1.8%	137394	1054	138214	232
	2 collection	0.5%	31320	93	31386	212
	3 collection	2.8%	41298	551	41752	207

and also the number of the formal rules. For example, there were 750 formal rules in the system developed by P. Kasparaitis, and in our system – only 400 rules.

5. Conclusions

1. The automatic system for transcription of Lithuanian language text which was based on the formal pronunciation rules and dictionary was created.
2. It was shown, when transcription of Lithuanian text is made not only with formal pronunciation rules, but also with the dictionary, errors reduced for
   - broadcast news text, from 2,16% to 0,93%,
   - frequency nonfiction literature text wordbook, from 1,42% to 0,8% errors,
   - medical scientific text, from 5,16% to 2,12%.
3. There was made a comparison between systems presented in this paper and created by (Kasparaitis, 1999), and results were
   - for broadcast news text, 1,8% errors for P. Kasparaitis and 0,93% – for our system,
   - for frequency nonfiction literature text wordbook, 0,5% errors for P. Kasparaitis and 0,8% – for our system,
   - for medical scientific text, 2,8% errors for P. Kasparaitis and 2,12% – for our system.

References

Ambrazas, V., *et al.* (1994). *Grammar of Modern Lithuanian Language*, Science & Encyclopedia Publishing Institute, Vilnius.

Baldwin, T., H. Tanaka (2000). A comparative study of unsupervised grapheme-phoneme alignment methods. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society (CogSci 2000)*, Philadelphia, USA. pp. 597–602.

Bentley, J., and R. Sedgewick (1997). Fast algorithms for sorting and searching strings, In *Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM Press.

Caseiro, D., I. Trancoso, L. Oliveira and C. Viana (2002). Grapheme-to-phone using finite state transducers, In *Proc. 2002 IEEE Workshop on Speech Synthesis*, Santa Monica, CA, USA.

Decadt, B., J. Duchateau, W. Daelemans and P. Wambacq (2001). Phoneme-to-grapheme conversion for out-of-vocabulary words in large vocabulary speech recognition, In *Proc. IEEE "Automatic Speech Recognition and Understanding Workshop*, Madonna di Campiglio, Italy.

Demuynck, K., T. Laureys, S. Gillis (2002). Automatic generation of phonetic transcriptions for large speech corpora. *ICSLP*, 333–336.

Dictionary of the Lithuanian Language (2005). I–XX, 1941–2002, electronic version, editorial board: G. Naktinienė (editor-in-chief), J. Paulauskas, R. Petrokienė, V. Vitkauskas, J. Zabarskaitė, Vilnius, Institute of the Lithuanian Language.
`http://www.lkz.lt`

Journal *Health Sciences. Public Health. Medicine. Nursing* (2005).

Karttunen, L. (1997). The replace operator. In E. Roche and Y. Schabes (Eds.), *Finite State Language Processing*. pp. 115–147.

Kasparaitis, P. (1999). Transcribing of the Lithuanian text using formal rules. *Informatica*, **10**(4), 367–376.

Keinys, S., *et al.* (2000). *The Dictionary of Modern Lithuanian Language*, 4th edition, Science & Encyclopedia Publishing Institute, Vilnius.

Libossek, M., F. Schiel (2000). Syllable-based text-to-phoneme conversion for german. *ICSLP*, **2**, 283–286.

Marcinkevičienė, R., A. Bielinskienė, V. Daudaravičius, E. Rimkutė (2004). Corpora for Lithuanian language technologies. In *Proceedings of the First Baltic Conference Human Language Technologies*, The Baltic Perspective, Riga, Latvia. pp. 21–24.

Marcinkevičienė, R. (2003). Abbreviations – acronyms – new words, *Kalbos kultūra*, **76**, 89–93.

National Committee of Lithuanian Language and Institute of Mathematics and Informatics, Vilnius, MII (2005). *Terminology of Lithuanian Language*.
`http://www.terminynas.lt`

Quazza, S., L. Donetti, L. Moisa, P. Salza (2001). Actor®: A multilingual unit-selection speech synthesis system. *Proc. of 4th ISCA Tutorial and Research Workshop on Speech Synthesis*.

Roche, E., and Y. Schabes (Eds.) (1997). *Finite-State Language Processing*, MIT Press, Cambridge.

Stoianov, P.I. (2001). *Connectionist Lexical Processing*. Doctoral Thesis at Groningen University.

Tillmann, C., and H. Ney (2000). Word re-ordering and DP-based search in statistical machine translation. In *Proceedings of the COLING 2000: The 18th International Conference on Computational Linguistics*, Saarbrücken, Germany.

Torstensson, N. (2002). Grapheme-to-phoneme conversion, a knowledge-based approach, *Fonetik*, **44**.

Vaitkevičiūtė, V. (2001a). *Pronunciation Background and Dictionary of Lithuanian Language*, Vilnius.

Vaitkevičiūtė, V. (2001b). *Dictionary of the International Words*, Vilnius, Žodynas.

Uneson, M. (2005). *Letter-to-Sound Conversion for Swedish with FST and TBL*.

Zhang, Z., M. Chu, E. Chang (2002). An efficient way to learn rules for grapheme-to-phoneme conversion in Chinese. *ISCSLP*, Taipei.

M. Skripkauskas has received master degree in applied informatics from the Vytautas Magnus University, in 2002. Now he is a graduate student at the Institute of Mathematics and Informatics. His main research interests are feature extraction of the speech signals and design of the automatic speech recognition systems.

L. Telksnys is a professor, doctor hibilitatis in informatics, Doctor Honoris Causa of the Kaunas University of Technology, a head of the Recognition Processes Department at the Institute of Mathematics and Informatics. He is the author of an original theory of detecting changes in random processes and the developer of computerized system for statistical analysis and recognition of random signals. His current research interests are: recognition of random processes, speech processing and computerized multimedia systems.

Lietuviškų tekstų transkribavimas naudojant žodyną

Mantas SKRIPKAUSKAS, Laimutis TELKSNYS

Aprašomas lietuviško teksto transkribavimas į fonemas šnekos atpažinimo reikmėms. Teksto transkribavimas į fonemas įgyvendinamas pasitelkiant formalias taisykles ir žodyną. Formalios taisyklės naudojamos ryšių nustatymui tarp teksto segmentų ir formalizuotų šnekos garsų – fonemų. Žodynas pasitelkiamas transkribavimo koregavimui bei žodžio kirčio ženklo ir vietos nustatymui. Automatinio transkribavimo kokybė patikrinta lyginant ją su rankiniu būdu gautais transkribavimo rezultatais. Parodyta eksperimentais, kad žodžių transkribavimo klaidų kiekis mažesnis negu 6%.