AUTOMATIC STRESSING OF LITHUANIAN TEXT USING DECISION TREES

Tomas Anbinderis

Department of Computer Science, Vilnius University
Naugarduko Str. 24, LT-03225 Vilnius, Lithuania
e-mail: Tomas.Anbinderis@mif.vu.lt

Abstract. This paper deals with one of the speech synthesizer components – automatic stressing of a text. The method, which by means of a decision tree finds sequences of letters that unambiguously define the word stressing, was applied to stress a Lithuanian text. Stressing rules based on sequences of letters at the beginnings, endings and in the middle of a word have been formulated. Also, the proposed method was compared with the method based on the morphological analysis, and it was proved that both methods gave similar results. The algorithm proposed in the paper reaches the accuracy of about 95.5%.

Keywords: text stressing, text-to-speech synthesis, decision tree.

1. Introduction

This paper deals with one of the speech synthesizer components – automatic text stressing. The speech synthesizer is a computer system, which can read any given text in a human voice. Speech synthesis can be divided into two main stages – linguistic processing and speech making. The linguistic processing stage creates the phonetic transcription of the given text, and is responsible for a necessary intonation and the durations of sounds (called prosody). The speech making stage converts obtained symbolic information to a human speech. This paper is concerned with one of the components of the linguistic processing stage – stressing (emphasis of one syllable with respect to others [6]). It should also be mentioned that determination of a word stress position and accent type can also be used to stress a text given to the news announcer, to teach stressing etc. [8].

Text stressing depends on the language. According to the stressing paradigm, languages can have a free stressing, or a fixed stressing. The location of a **fixed stress** can be defined by strict phonetic and phonological rules (for example, the first syllable, the penultimate syllable etc.). Latvian, Czech, Slovak, Icelandic, Estonian, Finnish, Hungarian, French, Polish languages have fixed stressing, In the case of a **free stress,** there are neither phonetic nor phonological rules, which determine how many syllables can precede or follow the stressed syllable. For example, the English, Romanian, Lithuanian, Slovenian, Russian, Italian and Spanish languages have free stressing [6]. In the case of a fixed stress, the stressing algorithm is usually defined by simple stressing rules and exceptions. In the case of a free stress, stressing methods and their complexity depend on the fact whether the language is inflectional or non-inflectional. Languages with free stressing only will be considered below.

Words of **non-inflectional** (with a low degree of inflection) languages (e.g., English) have few grammatical forms. Meanwhile words of **inflectional** (with a high degree of inflection) languages (e.g., Lithuanian, Russian) have different forms depending on the gender, number, case, degree, mood, tense, person, etc., and each form of the same word can have a different stress location.

If the language is **non-inflectional**, it is simply possible to build a vocabulary of all words with stresses. However, it is clear that the problem arises when stressing new words that are not included in the vocabulary (e.g., surnames). In this case one has to apply rules [8]. It should be noted that methods based on different rules [12, 4, 5] or even artificial neural networks [3] are often applied to non-inflectional languages too. **Inflectional languages** usually have no large databases that could specify correspondence between the spelling and pronunciation for all word forms [17]. This means that building a vocabulary for inflectional languages is a difficult task to perform. Therefore, methods based on morphological word inflection rules are most often used for inflectional languages. The Lithuanian language belongs to the group of inflectional language too. The Lithuanian language stress problem by means of morphological rules has been dealt with in several works already [7, 9, 10, 11, 13].

It should also be mentioned that the Lithuanian language stressing is made more complicated by additional stress elements, called **accents** (circumflex and acute). Quite a number of languages and dialects (e.g., Latvian, Slovenian) have one or another system of accents [6].

Thus, non-inflectional languages with free stressing use, as a rule, methods based on lists of stressed words, whereas free stressed inflectional languages – methods based on morphological words inflection rules. In this paper, the method which does not use any information about word forming morphemes, inflection, part of speech tags, boundaries of syllables etc. was applied for free stressed inflectional Lithuanian language. Methods applied to the other inflectional languages (Romanian [14], Slovenian [16]) usually use the information mentioned above. The proposed method uses a decision tree to find the sequence of letters, which unambiguously defines the word stressing. It appears that this method gives similar results as methods based on the morphological analysis. In the decision tree method the stressing rules are created automatically provided a sufficient quantity of stressed texts is available. The stressing algorithm is extremely simple, fast, can be easily adapted to other world languages, and easily ported to other programming languages and operating systems. Stressing rules based on sequences of letters at the beginnings, endings and in the middle of a word have been formulated. The algorithm reaches an accuracy of about 95.5%.

2. Stressed Texts Preparation

A significant quantity of stressed texts is needed seeking to create the stressing rules. Automatic stressing algorithm based on morphological analysis [7] was used for texts stressing. This algorithm was implemented in a special program that stresses text, marks out with a different color unstressed words and words that can be stressed in several ways, and allows the user to choose one stressing option or correct (add) the stress mark. Using this program, a professional philologist stressed and reviewed a set of texts containing about one million words (985967 words).

The texts were collected from the Internet and divided into six categories according to the genre: fiction, scientific literature, laws, republican periodicals, local periodicals, specialized and popular periodicals. When selecting texts according to the genre, the proportions of VDU corpus (*http://donelaitis.vdu.lt*, viewed as of 23 October 2008) were taken into account. Following the same proportions, texts were divided into five almost equal parts. See Section 6 how these parts were distributed to the creation, and testing of the rules.

3. Lists of Words

Before creating decision trees and stressing rules, we prepared two lists of words: unstressed and stressed word lists. **The list of unstressed words** is mainly made of clitics – words that tend to be unstressed in Lithuanian, e.g., the words “be” (without), “ant” (on), “bei” (and), “nuo” (from), “ir” (and) etc. This list also includes foreign words and abbreviations (for more on detecting clitics in Lithuanian see [1]). If some word appears in the text both as stressed and unstressed, it is put on the list of unstressed words in case it is more often unstressed than stressed. Then, the list of unstressed words (clitics) is always used before applying stressing rules, i.e. if the word under consideration belongs to this list, it is left unstressed and no stressing rules are applied to it.

The list of stressed words consists of the stressed words of the text. If the same words have different stressing (**homographs**), the stressing variant, which statistically occurs more frequently, is included in the list. The problem of homograph stressing in the Lithuanian language is considered in more detail in [2]. The list of stressed words is further used to make decision trees and stressing rules. The list can also be used as stressing rules but in this case it is difficult to expect good results, because the words that were not included in the training corpus, will remain unstressed (see Section 6).

4. Algorithm of Word Beginnings and Endings

As has already been mentioned, methods for drawing up stressing rules presented in the paper use classification (or decision) trees. Decision trees are used to forecast the variable *y* value, which corresponds to the parameter vector **f** (for more on creating and applying decision trees see [15]). In the present paper the parameter vector **f** corresponds to the sequence of letters in a word, and the variable *y* corresponds to the stressed letter (index) and the accent type. The essence of the method is to single out such letter sequences that define a unique word stressing. In the course of work three different methods were tested: letter sequences at the beginning, the end of a word and in any part of the word.

Let us first of all consider making of the tree taking letters from the beginning of a word (left to right). The tree nodes store the possible stressing (the index of the stressed letter and accent type), and the edges connecting the nodes store letters (Figure 1). We start with an empty tree, which has only one node – the root. Every word from the list of stressed words is added to the tree from the root starting with the first letter and working to the end of the word. When adding a word to the tree, all nodes that are on the path (i.e. as many as there are letters in the word) are complemented with the same information about stressing. After all words from the list are added to the tree, the root node stores all possible stressing variations. Below the algorithm is given in pseudo-code:

```
For each word from the list of stressed words
    The root becomes the current node
            For each letter starting with the beginning of the word
                Add a letter to the tree
                Change the current node and supplement it with information about stressing
```

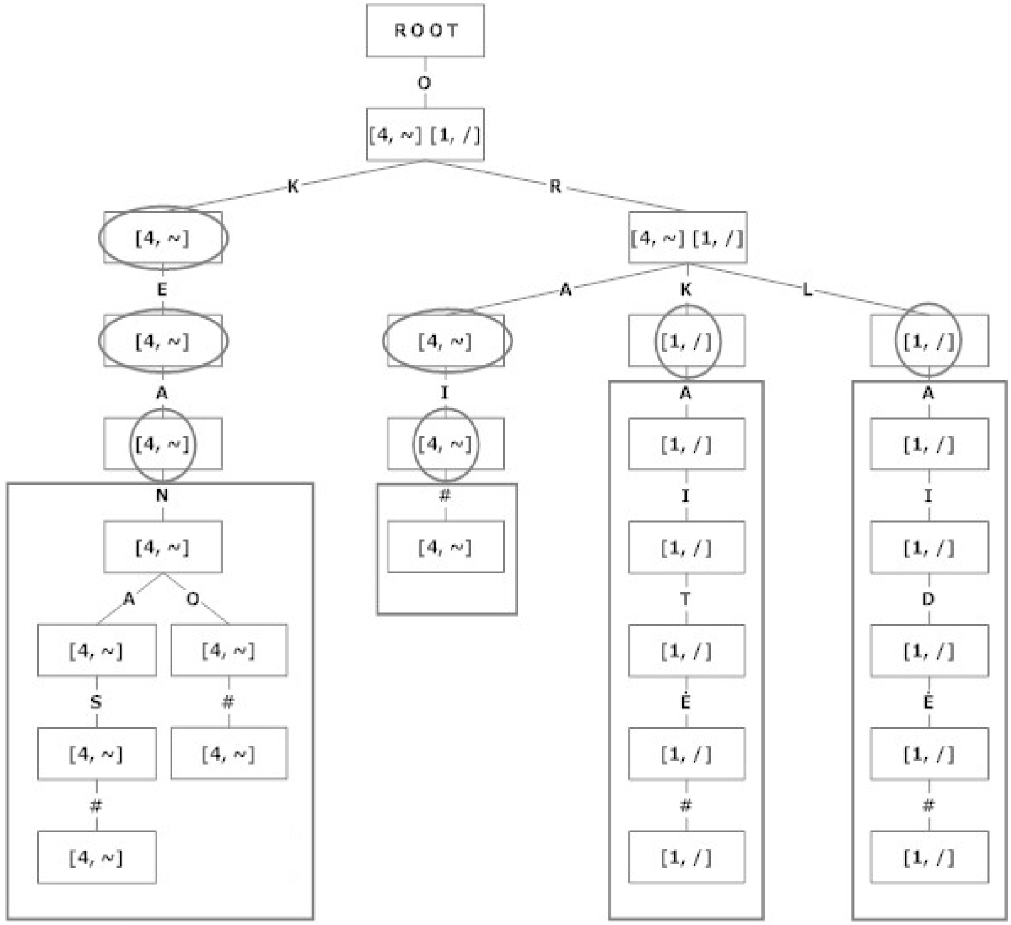

Figure 1. Decision tree of word beginnings. Circles indicate the decision nodes; rectangles show children of the decision nodes, which are not checked; ovals denote those nodes, which cannot be decision nodes because the stressed letter has not been reached

If one word is part of another word and these words have different stressing, two different stressing rules are to be formulated, but according to the proposed algorithm, only stressing rule corresponding with a longer word will be made. To solve this problem special word ending symbols ("#") were added to the end of every word. Following this operation the first (shorter) word is no longer a part of another word.

After adding all the words to the tree, it is possible to create the **rules** from it. Rules are sequences of letters that unambiguously define a word stressing. When moving from the root along all edges of the tree, we look for the nodes with a unique stressing, and where the index of the stressed letter is smaller than or equal to the level of the current node (referred to as **decision nodes**). The stressing rule is formulated by collecting all letters on the path into a sequence. It should be noted that children of the decision nodes could be skipped unchecked, because only longer rules defining the same stressing can be obtained from them. Only the shortest rule is added to the list of rules, for example, **ÓRK**, ÓRKA, ÓRKAI, ÓRKAIT, ÓRKAITĖ, ÓRKAITĖ#. Here and hereinafter a textual representation of the rule (i.e., for example, "ORK"), the index of the stressed letter (1) and the accent ('/') are combined into single form ("ÓRK").

Figure 1 and Table 1 show an example of a tree and rules created from five words. A list of words is presented in the left column of Table 1, and Figure 1 represents a decision tree created from these words (words are taken from the beginning). The rules formulated are given in the right column of Table 1.

Table 1. List of words and rules of word beginnings formed from them.

Input words	Beginning rule formed
OKEÃNAS# OKEÃNO# ORAĨ# ÓRKAITĖ# ÓRLAIDĖ#	OKEÃ ORAĨ ÓRK ÓRL

To stress a word with the word beginning rules all that is necessary to do is to find the rule that corresponds with the beginning of the word. To make a search faster, we can sort out the rules and make a binary search. If a suitable rule is not found the word is left unstressed. Usually, in such cases, the decision that is one level higher in the decision tree is taken, but in this paper the word is left unstressed (we can say that there exists another edge (that corresponds to all other letters) and its stressing decision is to leave the given word unstressed).

The tree of word endings is formed in the same way as the tree of word beginnings; only each word shall be reversed before it is added to the tree. Also, symbols of the word beginning ("#") rather than those of the word ending are added. In this case the root of the tree corresponds to the word ending, and that is why the rule is compared to the word ending when stressing.

5. Algorithm of the Word Middle Rules

Now we shall consider letter sequences, which can be anywhere in a word: at the beginning, in the middle and at the end. For the sake of simplicity, let us call them the word middle rules. The algorithm is similar to that of the word beginning, only each word is added to the tree several times (as many times as there are letters in the word) cutting one letter from the word beginning. It should also be noted that unlike the creation of the tree of word beginnings and endings, the additional symbol "#" is to be added to both the word beginning and its ending. Stressing rules are obtained from the decision tree in the same way as in the case of word beginnings. To stress the word, it is necessary to compare the rules not only with the word beginning but also to search for any part of the word that matches the rule. This slows down the search; therefore the issue of decreasing the number of rules becomes important.

As it has been previously shown (Section 4), we can reject the rule coinciding with the start of another rule. However, the list of the word middle rules also contains the rules that coincide with the ending of another rule, for example: #ORAĨ, ORAĨ, **RAĨ**. Starting a search with the longest rules, pairs of such of rules are found and longer rules are discarded. In the text below, this reduction of rules is called the **first reduction**.

Even after reducing the number of rules there might still be cases where several rules (determining the same stressing) suit the same word. Moreover, rules might differ in their statistical frequency of application. This must also be taken into account when reducing the number of rules. The main idea of the rule rejection algorithm is as follows: all words from which the rules were created are taken, and these rules are applied as long as all words become stressed. After the rules have been found for all words, the remaining rules can be deleted from the list. The rules are applied starting with those that suit the maximum number of words. In the text below, this reduction of rules is referred to as the **second reduction**. The second reduction in pseudo-code is as follows:

```
For each word determine which rules can be applied to it
For each rule calculate how many words it can be applied to
As long as at least one word is active
    Select the rule with a maximum number of words to apply
    Deactivate the words this rule can be applied to
    For each rule, that can be applied to the deactivated words reduce the applicable words counter
```

6. Experimental Results

As it was mentioned in Section 2, the available stressed texts were divided into five roughly equal parts containing 200000 words each. First of all experiments in which the same texts were used to create and test the rules were carried out. The experiments allowed the influence that homographs (words, which can be stressed in several ways) have on the stressing accuracy to be evaluated. During the experiments, when increasing the number of words from 200000 to 1000000, the error increased monotonically from 1.02% to 1.22%. Furthermore, the error rate did not depend on the method used to create stressing rules. This testifies to the fact that all proposed sets of stressing rules contain no less information than does the list of stressed words.

In further experiments some data were used to create the rules and other data were used to test them. In the present paper the data used to create the rules are referred to as **training data**, and the process of creating the rules is referred to as **training**. Data sets containing 200000, 400000, 600000 and 800000 words were used for training. These sets were obtained by combining corpus parts containing 200000 words in all possible ways (a total of 75 such combinations). Testing was conducted with all the words that were not used for training. The average error and the average number of rules were calculated for each training data quantity.

Experiments, using seven methods for creating stressing rules, were performed:

1. The list of stressed words was used as stressing rules (abbreviated **wrd**);
2. First, the word beginning rules were applied, then the word ending rules were used for the remaining unstressed words (**bgn-end**);
3. First, the word ending rules were applied, then the word beginning rules were used for the remaining unstressed words (**end-bgn**);
4. Only the word beginning rules were applied (**bgn**);
5. Only the word ending rules were applied (**end**);
6. A full set of the word middle rules (**mid**) or this set after the first reduction was applied (**mid1**);
7. A set of the word middle rules after the second reduction was applied (**mid2**).

Approach Six combines two methods because they give the same error (the only difference is the number of rules). It is worth mentioning that before stressing a word, first of all we check whether this word belongs to the list of unstressed words (clitics). If it does – the word is left unstressed, and no rules are applied to it. This method may also erroneously leave a certain number of words unstressed.

Averages of the text stressing accuracy for different methods for creating stressing rules are presented in Table 2. Here 800000 words for training and 200000 for testing were used.

As can be seen from Table 2, method 3 (**end-bgn**) gives the best result – 4.47% of error. Method 2 (**bgn-end**), method 6 (**mid**, **mid1**) and method 7 (**mid2**) give a somewhat greater (less than 0.3%) error. Methods 4 (**bgn**) and 5 (**end**), which use only the beginning or ending rules, give an error exceeding 6%. Finally, when the list of stressed words is used as stressing rules (**wrd**) the error exceeds 10%.

Averages of the number of rules for each method are presented in Table 3.

Table 2. Averages of the text stressing accuracy for different methods. Columns: A – clitics stressed (erroneous); B – clitics unstressed (correct); C – words unstressed (erroneous); D – unknown (e.g. foreign) words unstressed (correct); E – unknown (e.g. foreign) words stressed (erroneous); F – a wrong stress mark or stress place (erroneous); G – correct stressing; H – total errors (A+C+E+F); I – total correct (B+D+G)

Method	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
1 wrd	0.19	15.82	8.81	0.69	0.10	1.10	73.28	**10.21**	**89.79**
2 bgn-end	0.19	15.82	1.54	0.46	0.33	2.53	79.13	**4.59**	**95.41**
3 end-bgn	0.19	15.82	1.54	0.46	0.33	2.41	79.25	**4.47**	**95.53**
4 bgn	0.19	15.82	3.51	0.57	0.22	2.15	77.53	**6.08**	**93.92**
5 end	0.19	15.82	3.64	0.55	0.24	2.00	77.56	**6.07**	**93.93**
6 mid,mid1	0.19	15.82	1.04	0.35	0.44	2.99	79.17	**4.66**	**95.34**
7 mid2	0.19	15.82	1.93	0.44	0.35	2.29	78.98	**4.76**	**95.24**

Table 3. Averages of the number of rules for each method

Method	**200000**	**400000**	**600000**	**800000**
only clitics	1381	2441	3407	4309
wrd	43215	67564	86633	102760
bgn	28293	42608	53433	62424
end	29790	44688	56009	65404
mid	118442	175615	218280	253379
mid1	39545	57165	70047	80510
mid2	19627	28840	35676	41291

As can be seen from Table 3, following the second reduction (**mid2**), the word middle method requires the minimum number of rules. Beginning (**bgn**) and ending (**end**) methods requires about one and a half times more rules. Thus, the method that is best in respect of the error, (**end-bgn**) requires about three times more rules than the word middle method after the second reduction (**mid2**). However, the word middle method works much slower, because the rules must be compared not only with the beginning or ending of a word but also starting with each letter in a word. The second reduction of the word middle rules allows the number of rules to be reduced by as much as twofold (compared to (**mid1**)), while the accuracy decreases only by 0.1%. Another conclusion is that the number of the word ending rules (**end**) is always somewhat larger than that of the word beginning rules (**bgn**).

7. Coparison of Results with the Morphology-based Method

In Table 4 the best method (**end-bgn**) is compared with the method proposed in [7] [9] that is based on morphological rules. This method has been supplemented with clitics [1] and homographs [2] stressing rules. It is worth mentioning that this method was used for the initial preparation of data (see Section 2). Both methods are tested with identical data. In this case, the algorithm proposed in this paper was trained with only one set containing 800000 words, so the average was not calculated, consequently, the results of the algorithm differ from those presented in Table 2. Though the results of the method proposed in this paper are slightly worse than those of the morphological approach (about 0.8%), the proposed method is much simpler with respect to both the creation and application of the rules.

Table 4. The best method (**end-bgn**) as compared with a morphological approach (**morpholog**) testing with only one set containing 200000 words. Columns: see Table 2

Method	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
3 end-bgn	0.17	15.80	1.32	0.51	0.36	2.37	79.47	**4.22**	**95.78**
morpholog	0.07	13.49	1.54	3.07	0.11	1.67	80.05	**3.40**	**96.61**

8. Forecasting the Influence of the Training Data Size on the Stressing Accuracy

As could be expected, experiments showed that the larger number of words was used to create the rules, the greater accuracy of the new text stressing was achieved. On the basis of the error values obtained using 200000, 400000, 600000 and 800000 words for training the most accurate method (**end-bgn**), the attempt was made to forecast (extrapolate) an error for a greater number of training words. The method of the least squares (*http://www.wolfram.com*, viewed as of 1 July 2009) was used for extrapolation. Results are presented in Figure 2. The extrapolation function obtained is as follows: $y = 4.0256*x^{-0.4449}$, where x is the number of words, and y is the error percentage. The number of errors, similar to that achieved by means of morphological method (3.40%) would be achieved when training the method (**end-bgn**) with 1500000 words, whereas having trained the method (**end-bgn**) with 2000000 words the forecasted error accounts for 2.96%.

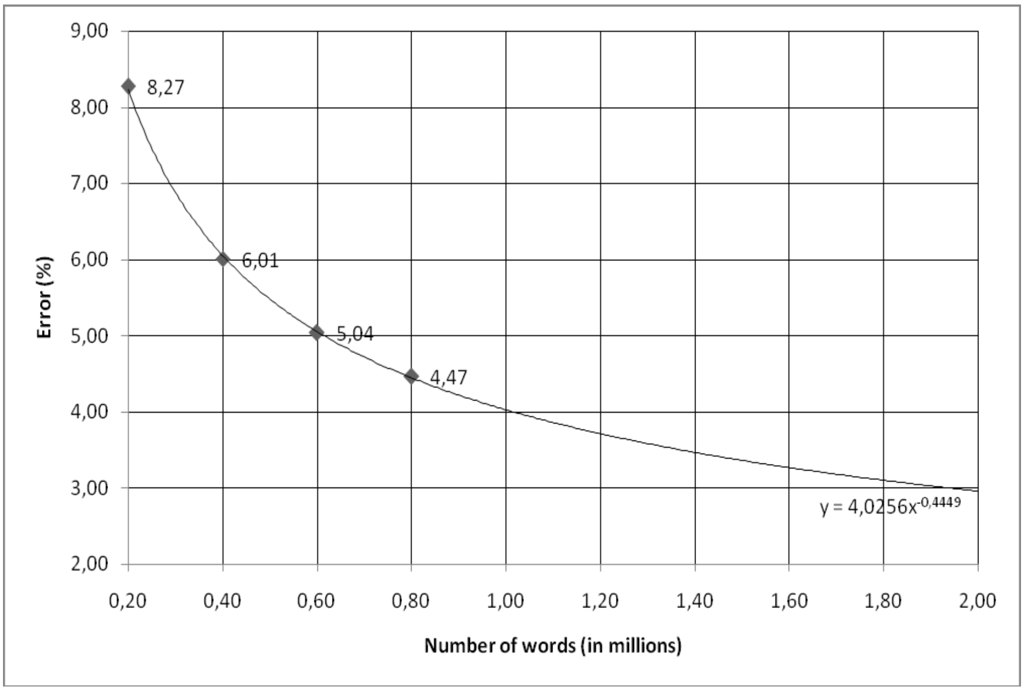

Figure 2. Forecast of an influence of the training data size on the stressing accuracy.

9. Conclusions

Methods to create stressing rules from the list of stressed words are offered in this paper. Such methods are usually applied to non-inflectional languages. The present paper shows that such a method can be successfully applied to the highly inflectional Lithuanian language too.

The decision tree algorithm was used to create the rules. Several methods for building decision trees were considered. We showed that the ending-beginning approach ensures the greatest accuracy, and by applying the word middle rules method the minimum number of rules is obtained (the number of rules was reduced by means of the algorithm described in this paper).

By its accuracy the proposed best method gives only a 0.8% greater error than the method based on morphology. However, it is shown that with an increase in the size of training data, this accuracy can be expected to be reached and improved.

It is worth mentioning that the proposed methods are based only on sequences of letters and do not require any knowledge of the language: morphemes, parts of speech, word inflection, syllable boundaries etc. They need only a sufficient number of stressed texts from which the rules will be created automatically. Therefore, these methods can easily be adapted to other languages.

The stressing algorithm itself is in essence a binary search in the sorted list of rules, therefore the algorithm is very fast, and it can easily be transferred into another programming language or another operating system. Portability is a great advantage of the proposed method as compared with the morphology-based approach.

References

[1] **T. Anbinderis, P. Kasparaitis.** Algorithms for Detecting Clitics in the Lithuanian Text. *Studies about Languages*, 10, 2007, 30-37 (*in Lithuanian*).

[2] **T. Anbinderis, P. Kasparaitis.** Disambiguation of Lithuanian Homographs Based on the Frequencies of Lexemes and Morphological Tags. *Studies about Languages*, 14, 2009, 25-31 (*in Lithuanian*).

[3] **J. Arciuli, J. Thompson.** Improving the Assignment of Lexical Stress in Text-to-Speech Systems. *Proceedings of the* 11*th Australian International Conference on Speech Science & Technology*, 2006, 296-300.

[4] **K. Church.** Stress Assignment in Letter to Sound Rules for Speech Synthesis. *Proceedings of the* 23*rd Annual Meeting on Association for Computational Linguistics*, 1985, 246-253.

[5] **K. Church.** Morphological Decomposition and Stress Assignment for Speech Synthesis. *Proceedings of the* 24*th Annual Meeting on Association for Computational Linguistics*, 1986, 156-164.

[6] **A. Girdenis.** Theoretical Foundations of Lithuanian Phonology, 2-nd edition. *Science & Encyclopedia Publishing Institute, Vilnius,* 2003 (in Lithuanian).

[7] **P. Kasparaitis.** Automatic Stressing of the Lithuanian Text on the Basis of a Dictionary. *Informatica*, 11(1), 2000, 19-40.

[8] **P. Kasparaitis.** Lithuanian Text-to-Speech Synthesis. Doctoral thesis. *Vilnius University, Vilnius*, 2001 (*in Lithuanian*).

[9] **P. Kasparaitis.** Automatic Stressing of the Lithuanian Nouns and Adjectives on the Basis of Rules. *Informatica*, 12(2), 2001, 315-336.

[10] **A. Kazlauskiene, G. Raskinis.** The Possibilities of an Automated Verb Stress. *Language Theory and Practice*, 2004, 80–82 (*in Lithuanian*).

[11] **A. Kazlauskiene, G. Norkevicius, G. Raskinis.** The Automatic Accentuation of Lithuanian Language Verbs: Related Problems and Their Resolution. *The Problems of Phonetics and Accentuation in Baltic and Other Languages*, 2004, 166–173 (*in Lithuanian*).

[12] **D. L. McPeters, A.L. Tharp.** Application of the Liberman-Prince Stress Rules to Computer Synthesized Speech. *Proceedings of the First Conference on Applied Natural Language Processing*, 1983, 192 – 197.

[13] **G. Norkevicius, A. Kazlauskiene, G. Raskinis.** Accentuation of Lithuanian Nouns and Adjectives: Structural Model, Algorithm and Implementation. *Studies about Languages*, 6, 2004, 72–76 (*in Lithuanian*).

[14] **E. Oancea, A. Badulescu.** Stressed Syllable Determination for Romanian Words within Speech Synthesis Applications. *International Journal of Speech Technology*, 5(3), 2002, 237-246.

[15] **J.R.Quinlan.** Induction of decision trees. *Machine Learning*, 1, 1986, 81-106.

[16] **T. Sef.** A Two Level Lexical Stress Assignment Model for Highly Inflected Slovenian Language. *Proceedings of the Third International Conference on Information Technology and Applications*, 2005, 347-351.

[17] **R.W. Sproat.** Multilingual Text-to-Speech Synthesis. *Kluwer Academic Publishers, Norwell, MA*, 1997.

Received November 2009.