EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 1, Issue 2(2013) 102–109

A "BY PART" METHOD OF RUSSIAN WORD SPEECH RECOGNITION

A.V. Nicenko

Abstract The present article is a description of a speech recognition method based on the idea of recognizing words by their component parts. The method proceeds from automatic phonetic segmentation, using full variation digital analogue, to further compose a diphone base and carry out a DTW algorithm-based speech recognition: firstly, for a variable word part (a quasiflexion) and secondly, for its static part (a quasibase), with reference templates automatically formed from diphone templates. It results in considerable reduction of the running time and the reliability growth of word form speech recognition. This method can be employed for recognizing large and very large vocabularies.

Key words: segmentation of speech signal, diphone, dynamic time warping, feature vector, quasiflexion

AMS Mathematics Subject Classification: 68T10, 68T50

1 Introduction

The paper develops the idea of recognizing words "by (a component) part" suggested in [1]. There are two tasks here: to enhance the validity of recognizing forms of the same word and to reduce the size of vocabularies under recognition.

A complicated mechanism of inflexion is a problem for Russian speech recognition technologies[2]. In contrast to English Russian is characterized by a rich variability of word forms. The correct word forms are indispensable for correct syntax ensuring the comprehension of speech flow. Russian orthographic dictionaries contain about 100 thousand words as language entries. Every entry, except grammar words, possess their own word forms' sets, and many, especially verbs, have rather large sets. Due to this fact a dictionary of Russian word forms can run into 3 million items. In terms of recognition different word forms make up different classes. This constitutes a great problem for Russian speech recognition: an enormous quantity of classes.

The recognition of the same word forms poses a harder task than the recognition of word forms for different words. It is caused by inflections, that are typically unstressed and subject to reduction in the spoken chain. To ensure the validity of word forms' recognition we suggest starting with in flexions. Our considerations in favor of this procedure are the following: the two forms of a longish word have a general base, hence have more common features than different, which happens to be a source of errors in recognition. Since inflections have more differences relative to whole word forms, their recognition must be less error-prone. On the other hand the DTW-recognition is better for longer speech segments. So it makes sense to add a part of a suffix to inflexions

and work on with these objects as quasiflexions. Logically, the rest of the word will have a name of a quasibase.

Using quasiflexions leads to downsizing the vocabularies under recognition. The quasiflexions are obviously common for large word lists. If we have m quasibases and n quasiflexions, their combinations will form $m \times n$ word forms. If we could recognize whole word forms we would have $m \times n$ recognition vocabulary, while recognizing quasibases and quasiflexions separately we have $m + n$ recognition objects. As a result the recognition time decreases considerably and the reliability of recognition grows.

Thus, to solve the problem mentioned above we suggest a two-step recognition procedure: firstly, to do the variable part of a word (a quasiflexion) and secondly, the stable part (a quasibase) from a set corresponding to a recognized quasiflexion.

2 Method description

Assuming that Russian is an inflectional language (i.e, syntactically controlled with the help of word forms, produced by using inflections), the words are modeled as combinations of fixed and variable components:

$$x = c\,(x)\,\&f\,(x) \tag{1}$$

where $c\,(x)$ is a part of lexeme x, which remains unchanged in the process of inflecting (quasibase), $f\,(x)$ is its variable component (quasiflexion) and & is a concatenation sign.

Considering sets of word forms of various words, their quasiflections may be combined into one vocabulary and the quasibases into another. Since the recognition will be performed using diphone reference templates (see [3]), every quasi-base and quasiflection will receive a phonetic transcription and a chain of relevant diphones. For example, for a word "vocalization":

vokaliza → vakaliza → v0-va-ak-ka-al-li-iz-za-a2

tsija → tsija → ts0-tsi-ij-ja-a2

To build a phonetic transcription we used a simplified automatic transcriptor, meeting the recognition requirements. It is implemented as a program, replacing some symbols with other symbols according to the rules contained in the control file. Each of these rules is written as two parts connected by an equality sign. On the left there are original characters corresponding to a word in writing, and on the right there are transcription characters replacing the original sequence. When transcribing a word, the program sequentially searches for the left side for an occurrence of a rule, and on finding one, replaces it with the right side counterpart.

We use 8-bit recording with sampling frequency 22050 Hz. The recorded speech input from the microphone is automatically segmented and labeled into discrete parts corresponding to broad phonetic transcription. The voiced units of the speech flow are isolated from a set of unvoiced units. To label the unvoiced units, the speech signal is blocked into non-overlapping frames of 256 samples to undergo a 100 to 200 Hz bandpass filtering. This procedure makes hissing sibilants and affricates ([s], [sh], [shch], [ts], [ch]) similar to unvoiced stops ([p], [k], [t]). After this step each 256-sample frame is classified as an unvoiced sound by measuring the number of constancy points

(that is, those discrete points in time whose value of the signal remains the same at the next moment) for each frame, which is higher for voiceless stops. Next, the voiced units are segmented and labeled as vowels or voiced consonants using a numerical analogue of total variation (see [4]). Thus a set of labeled segments, corresponding to four phonetic classes is formed: vowels (W), voiced consonants (C), fricatives (F) and pause-like segments (P), corresponding to the voiceless stop consonants or the stop part of an affricate (Fig. 1).

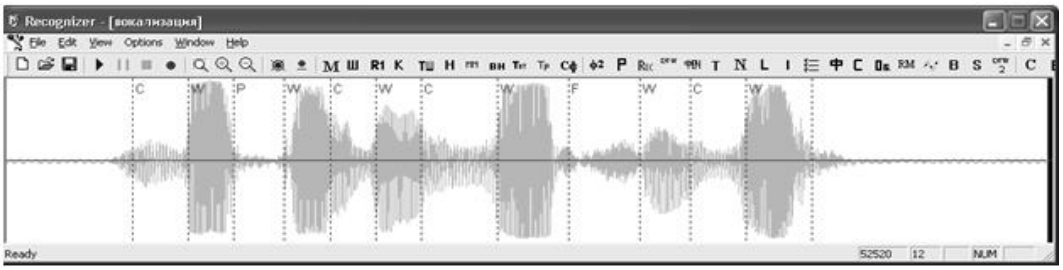


Figure 1: Segmentation and labeling for the word "vocalization".

A classical algorithm known as Dynamic Time Warping (DTW) is applied for recognition [5,6]. It uses feature vectors based on relative frequencies of the oscillation lengths, measured for 368-samples frames (see [7]). The reference templates for words in recognition vocabulary are compiled from diphone templates whose full base amounting to 1600 items is created for every speaker in advance. A base like this relieves the user from the need to create any voice-performed reference template.

Creating an automatic recognition system requires the formalization of the concepts used. Thereby, the concept of a T«truncated diphoneTï is introduced. By a "truncated diphone" we understand an interphonemic transition within a word, indicating a section of standard length: 3 frames of 386 readings to the left from the inter-sound label and 3 similar frames to the right of it. Subsequently, we will omit the word T«truncatedTï to denote this same object by the term "diphone". The diphone reference template is a set of 6 feature vectors, associated with a symbolic name for corresponding sounds. Besides, we use a section of 3 frames at the beginning of the word (named by the beginning phoneme followed by symbol "0") and a section of 3 frames in the end of the word (named by the end phoneme followed by symbol "2"), provisionally referring to them as initial or end semidiphones of a word (transition from silence to speech and vice versa). Since initial voiceless stops [k], [p], [t] and their palatalized counterparts are not labelled during segmentation, we also use "ka0", "pi0", "tl0", etc. as initial semidiphones.

Assume that we have a reference template of some spoken word, consisting of a set of 29-dimensional vectors:

$$E = (e_1, e_2, ..., e_m) \tag{2}$$

Such template is created for each word from the recognition vocabulary. Suppose now that

$$A = (a_1, a_2, ..., a_k) \tag{3}$$

is a test pattern of the word we need to recognize. To measure a similarity between two word patterns, which may vary in time or speed, we are using the method of dynamic programming [6].

To be specific, let's choose as the distance between two feature vectors the sum of absolute differences of the corresponding coordinates (l_1 - metric). Let's denote the distance between the vectors e_j and a_j of feature sequences (2) and (3) by D_{ij} and for every $1 \leq i \leq m$, $1 \leq j \leq k$ define the value C_{ij} as:

$$\begin{gathered} C_{11} = D_{11}, C_{i1} = D_{i1} + C_{(i-1,1)} \\ C_{1j} = D_{1j} + C_{(1,j-1)}, C_{ij} = D_{ij} + min\left(C_{(i-1,j)}, C_{(i,j-1)}, C_{(i-1,j-1)}\right) \\ 2 \leq i \leq m, 2 \leq j \leq k. \end{gathered} \tag{4}$$

C_{ij} is a local cost measure, that is proportional to the distance between the part of the signal represented by feature sequence (2), from the beginning to the i-th frame, and a signal represented by (3), from the beginning to the j-th frame. The DTW-distance between for two word patterns is defined as $\frac{C_{mk}}{\sqrt{(m^2+k^2)}}$, where C_{mk} is a total cost of a warping path. $\sqrt{(m^2+k^2)}$ is a path normalizing factor which is needed to normalize the distances for long and short words in the dictionary.

Evaluating the local cost measure C_{ij} for each pair of elements of the sequences E and A, one obtains the cost matrix. Then the goal is to find an alignment path between E and A having minimal overall cost among all possible warping paths. Intuitively, such an optimal alignment runs along a "valley" of low cost within the cost matrix C (Fig. 2).

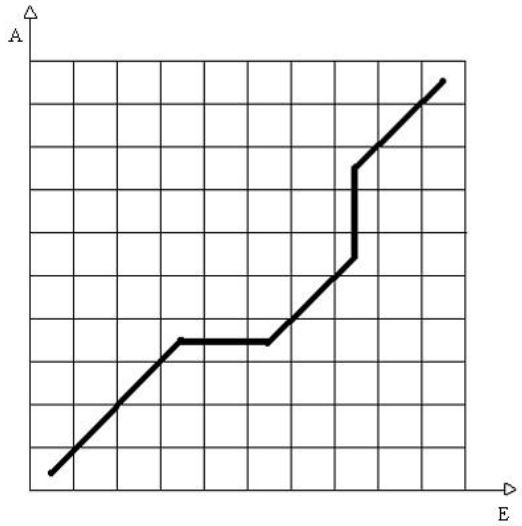


Figure 2: Illustration of the time alignment path for the feature sequences A and E.

All feature vectors in diphone patterns are represented as a code vectors and formed a codebook B. All diphone patterns and all feature vectors are indexed. Every word in recognition vocabulary is automatically transcribed, and by transcription a chain of diphone names is constructed. Each of them is represented the corresponding diphone

template. The resulting chain of vectors forms a word reference template [4]. All word patterns is stored in memory as a hierarchical tree structure, so it greatly speeds up the recognition process. Each diphone is represented in vocabulary tree with his index number, and each word pattern corresponds to some path within the tree. If some branches have a common part, the calculation, filling the relevant part of DTW-matrix, are performed only once.

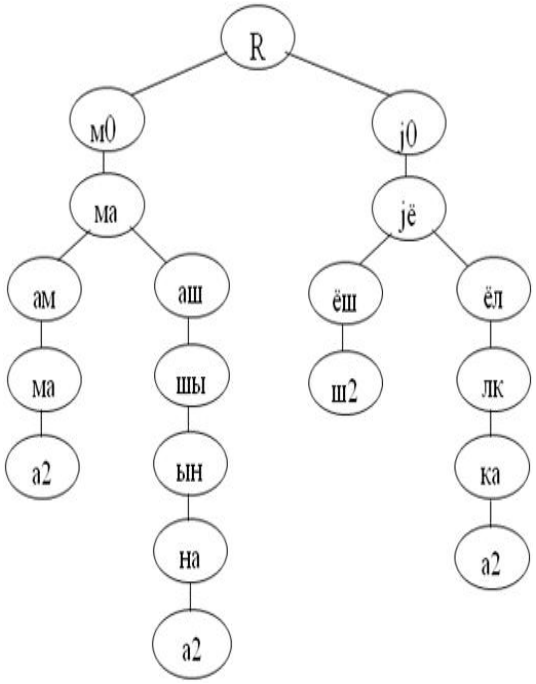


Figure 3: A recognition tree for simple vocabulary

Each level of the vocabulary tree match the diphone position in the word sample (Fig. 3). Each node in each level represents the index number of diphone, which is present in the word on the appropriate position. Node which represents the final diphone in word sample is marked as the end of the word sample (the index number of the corresponding word in the vocabulary is assigned with this node). If the node is not the word end it is assigned with the value of -1. The maximum depth of the tree corresponds to the maximum word length (in diphones) in a dictionary.

The recognition process starts with automatic segmentation and labeling, and then applying a so-called interphoneme processing to the input signal: removing stationary components of the phoneme units and left only diphones around phoneme labels (interphoneme transitions). Then a pattern of the processed input signal as a set of feature vectors is created and a global table D of distances from these vectors to all the codebook B vectors is calculated. Next, the DTW-distances is calculated from the input word pattern to all the word samples by recursively traverse the tree with depth-first preorder walk method. The walking starts from tree root, and then goes down the branches until they reach the last child, marked as the end of the word sample. Once the end of a word sample is reached, the post-order operation is performed along the traversed path until reaching the node, which has not yet visited, and then new branch

traversing starts. The process is completed when the root of the tree is reached, and all the nodes have proved to visit.

When traversing the tree, a chain of vector index numbers from the visited tree nodes forming a word sample. Once the end node is reached, the DTW-distance from chaining vectors (word sample) to the chain of input pattern vectors is calculated. The distances between the vectors are taken from global distance table D. During calculation the DTW-matrix does not rewriting completely. Only the columns that corresponds to the new code vector numbers added to the chain are updated.

Quasiflexion recognition is based on the principle of minimal DTW-distance [6]. The end part of a speech signal is sequentially recognized with DTW algorithm, starting from the two end segments: firstly, the last two consecutive segments are selected, then three, four, etc. until the predetermined maximum of phonetic segments (Fig. 4).

Thus, a list of hypothetical quasiflexions and DTW-distances between their patterns and the corresponding speech segments is created. Out of this list a quasiflexion with minimal distance is selected.

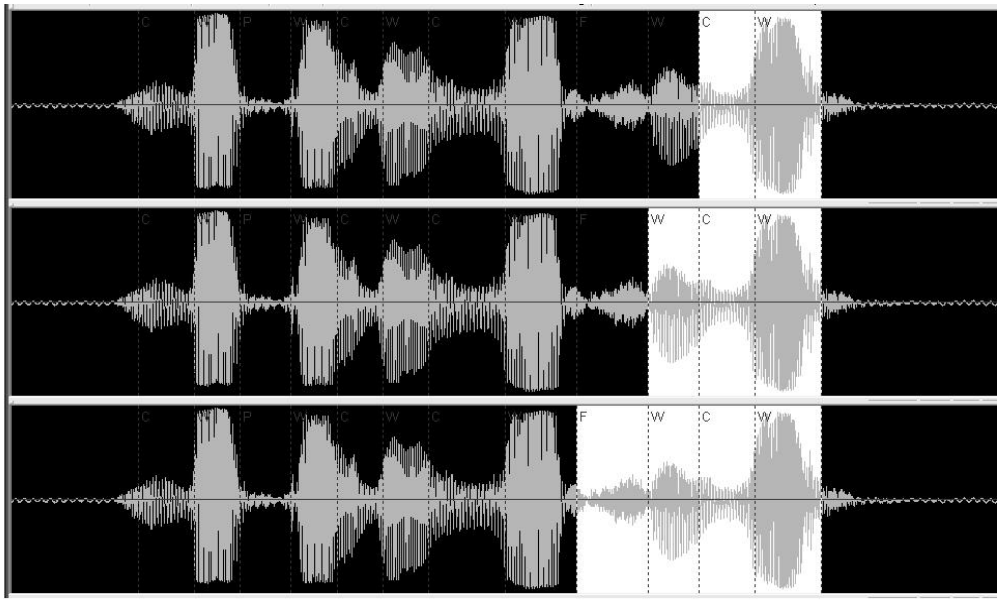
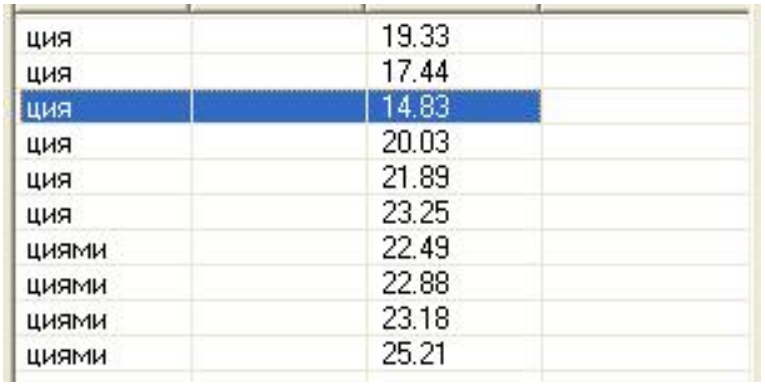


Figure 4: Segmentation and labeling for the word "vocalization".

Fig. 5 shows an example of such a list obtained during the recognition of the word "vocalization". As a result of ten sequential recognitions with a quasiflexion vocabulary a list of hypothetical quasiflexions is obtained and DTW-distance to them is calculated. Among them the "tsija" has the minimal distance, enabling us to identify it as the end part of the word under recognition.

Then, for the selected quasiflexion the appropriate quasibase vocabulary is accessed, and within its lexicon a DTW-recognition is performed for the beginner part of the word until the part corresponding to the detected quasiflexion.



ция		19.33
ция		17.44
ция		14.83
ция		20.03
ция		21.89
ция		23.25
циями		22.49
циями		22.88
циями		23.18
циями		25.21

Figure 5: A list of hypothetical quasiflexions for the word "vocalization".

3 Conclusions

This paper proposes a "by-part" method for Russian word speech recognition, based on the segmentation and dynamic time warping algorithm, and suited to the Russian inflexion system. It is promising since it can solve the difficulty of applying existing technologies of speech recognition to the Russian language, characterized by a complex mechanism of word formation. The method aims to reduce the computational cost in recognizing and to increase the reliability of the word forms' recognition. It can be used for recognizing large and very large vocabularies.

References

[1] V.Ju. Shelepov, A.V.Nicenko, H.V.Dorohina, *On some questions of Diphone recognition and the recognition of continuous speech*, Artifical Intelligence, Vol. 3 (2013), pp. 209-216.

[2] A.L. Ronzhin, R.M. Yusupov, I.V. Li, A.B. Leontieva, *Survey of Russian Speech Recognition Systems*, SPECOM'2006, St. Petersburg, 2006, pp. 54-60.

[3] V.Ju. Shelepov, A.V.Nicenko, H.V.Dorohina, *On speech recognition using phoneme transition base*, Artifical Intelligence, Vol. 1 (2012), pp. 132-139.

[4] V.Ju. Shelepov, A.V.Zhuk, A.V.Nicenko, *Computer voice control system development on example of mathematical formula voice input task*, Artifical Intelligence, 3, 2010, pp. 259-267.

[5] T.K. Vintsyuk, *Analiz, raspoznavanie i interpretatsiya rechevyh signalov*, Kiev: Naukova dumka, 1987.

[6] M.K. Brown, L. Rabiner, *An adaptive, ordered, graph search technique for dynamic time warping for isolated word recognition*,IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 30, Issue 4 (1982), pp. 535-544.

[7] V.Ju. Shelepov, *Speech recognition lectures*, Donetsk:IPII, Nauka i osvita, 2009.

[8] V.Ju. Shelepov, A.V.Nicenko, *On the Problem of Continuous Speech Recognition*, Artifical Intelligence, Vol. 4 (2012), pp. 272-281.

A.V. Nicenko,
Institute of Artifical Intelligence,
118-b, Artyom st, 83048 Donetsk, Ukraine
Email: `nav_box@mail.ru`,
Received 2 Dec 2013, Accepted 17 Dec 2013