Interpretation of continuous Ukrainian pronunciation for spoken dictionary-interpreter

Taras Vintsiuk, Mykola Sazhok, Valentyna Yatsenko
International Scientific Educational Center of Informational Technologies and Systems
40 prospekt Akademika Hlushkova, Kyiv, 03680

Abstract

A problem of continuous speech interpretation within a subject domain is considered. A way to specify allowed sequences of words in phrases by means of LISP-structures is considered in frames of the generative model for speech understanding for highly inflective languages with relatively free word order. Experimental research presents promising results, which allowed creating a spoken vocabulary-interpreter prototype.

1. Introduction

The problem of accuracy improvement for speech signal recognition and understanding is topical to date. One of the possible ways to reach this aim is an efficient specification of different restrictions (syntactical and semantic) for permissible sequences of words in phrases and their modeling in automatic speech understanding algorithms.

Where it is needed? It is important for different systems of spoken dialog: interpreter, phone services etc.

Developed earlier automatic modules for Slavic languages provide quite exact sentence recognition [1] but still requires huge arithmetical resources. These modules also don't operate with multi-choice cases. It happens due to difference between Slavic languages and i.e. English. The main difference is huge amount of word forms and relatively free word order. These particularities significantly complicate the task of speech signal recognition and understanding.

The consequent problems are setting of all possible variants of the dialog language, which render the same contest, as well as a problem of generation and search for the most relevant signals as well as working out restrictions on sequences of words in phrases according to the structures that specify sentences.

For analysis and working out the restrictions on allowed sequences of words in phrases it was proposed to consider LISP-structures [2, 3]. Huge amount of sentences with the same meaning are generated basing on these structures. To increase the effectiveness of recognition algorithms it was proposed to automate the building of LISP- structures.

In chapter 1 we present the general characteristic of the recognition problems and contest interpretation, in the second chapter we present the task setting concerning sentence specification taking into account the restrictions on allowed sequences of words, third chapter contains experimental results.

2. General characteristic of the recognition tasks and continuous speech interpretation

Lets look into the essence and correlation of recognition tasks and continuous speech interpretation task [2, 3]. Speech recognition is the process of automatic processing of the signal with the aim to define the word order expressed by this signal.

Speech understanding is the process of automatic processing of speech signal with the aim to define the content and to present this content in the canonic form easy for further usage. It is obvious that speech understanding is higher level of information summarizing as one and the same meaning could be rendered in some words constructions. To obtain the better results of recognition and understanding these tasks should be performed in the one interrelated process.

As each thought or statement could be expressed by different sentences with the same meaning we should define the certain restrictions on words constructions in the sentences. That is why while interpretation of the contest the different sentences with the same meaning should be reflected in the one result, i.e the result should not contradict to syntax, semantics and pragmatics of the subject area. Taking the above said into account it is proposed to examine the models of signals recognition considering syntax and semantic [2, 3].

The task of speech understanding is more complicated then recognition as for its settling it is necessary to use additionally the a priori information. That is why the first thing is to learn how to set economically all possible sentences in the dialog language. There are different ways. One way is to build LISP-like structures and with their help to define the restrictions on allowed sequences of words.

This method of continuous speech recognition and understanding could be realized in the form of the generative model of understanding of continuous speech [2].

According to this model the main task is the recognition of contest expression among the specified set of contest expressions. It is necessary to indicate which contest expression from the set is actually contained in the speech signal.

Each contest expression we give in the canonic form which is written in a certain semantic language (formal arithmetical language rendering notions and their relations). Then, using the Generator of Semantic Equivalents (GSE) we assign the transformation of the canonic form which does not infringe the contest. In this way GSE generate all possible sentences with the same meaning defined by the canonic form. Then we introduce transformations which generate all possible etalon signals of the continuous speech for each sentence generated by GSE. These etalon signals are different in tempo (changed nonlinear) and in intensity of pronouncing.
0100090000035959000009003a000000000014000000026060f001e00ffffffff040014000000576f72640e004d6

Figure 1. Model of the synthesis of the etalon signals for content interpretation.

Further it will be examined variants of setting of all possible sentences with the same meaning, generation and search of the most relevant signals and working out the restrictions on allowed sequences of words according to the structures proper for a sentence.

It is given that tasks of speech recognition and understanding should be settled in the correlated process when the recognition is lead from the semantic-syntactic side when the highest reliability of recognition is achieved.

3. Taking into account the restrictions on allowed sequences of words

Within the generative model for recognition of the speech signals it is proposed to examine the certain hierarchy of sentences` order. It means that all sentences of the language we divide on subject areas (SA) like in a phrase book. Each SA consists of a certain number of Meaning Category (MC). For example, Restaurant SA consists of such MC: booking the table, menu, ordering etc. There are not so many MC for each SA. In each MC there are a number of equivalent Sentence Types (ST) that are specified by LISP-structures [2]. ST – is a construction that economically specifies a set of sentences obtained by independent substitutions and inversions of certain words and words combinations.

Let’s examine an ST “asking for help” of the respective MC for the SA “Everyday Phrases”.

$$\left[\begin{matrix}\textit{Чи}\\ \textit{Whether}\\ *\end{matrix}\right]\left(\left(\left[\begin{matrix}\textit{не}\\ \textit{will not}\\ *\\ \textit{will}\end{matrix}\right]\ \underset{\textit{help}}{\textit{допоможете}}\right)\left(\left[\begin{matrix}\textit{Ви}\\ \textit{you}\\ *\end{matrix}\right]\left[\begin{matrix}\textit{мені}\\ \textit{me}\\ *\end{matrix}\right]\right)\left(\begin{matrix}\textit{вирішити}\\ \textit{to fix}\\ \textit{розв`язати}\\ \textit{to solve}\end{matrix}\right)\left(\underset{\textit{this problem}}{\textit{цю проблему}}\right)\right)$$

The parentheses () contain invertible subdictionaries, and square brackets [] contain non-invertible subdictionaries. Subdictionaries can be inverted only within “superior” brackets. The * symbol - means an empty word.

It is not difficult to assure that this example set $2 \cdot 4! \cdot 2 \cdot 4 \cdot 2 \cdot 1 = 768$ of different sentences with the same meaning permissible in Spoken Ukrainian. Among them are the following sentences:

Чи цю проблему не допоможете вирішити Ви мені?
Whether this problem will not help to fix you me?
Ви мені цю проблему вирішити допоможете?
Will you me this problem to fix help?

Thus we can see that in this ST there are a lot of syntactically possible sentences of the spoken language. To build all possible sentences of spoken dialog we use a so-called Oriented Semantic Network (OSN) [2].

To build an OSN we will use the mentioned above MC and ST. The main element of the ST is a subdictionary. The subdictionaries are named depending on their belonging to SA.

The OSN has states which we indicate with "y". Among them: the first one y_{start} and the final one y_{end}. States $y = \mu$ and $y = \nu$ are connected with arrows. The subdictionary $z_{\mu\nu}$ is ascribed to each arrow. Moving along the arrow $\mu\nu$ which connects states μ and ν we will choose only one word from the subdictionary $k : k \in z_{\mu\nu}$.

We will build the OSN in such a way during which while moving from y_{start} to y_{end} only permissible sentences were created i.e. the sentences, which meet the requirements of syntax, semantic and pragmatic of SA. It is desired the OSN to be built with the less possible states.

The OSN for the sentence from the above-mentioned example is shown in Figure 2.

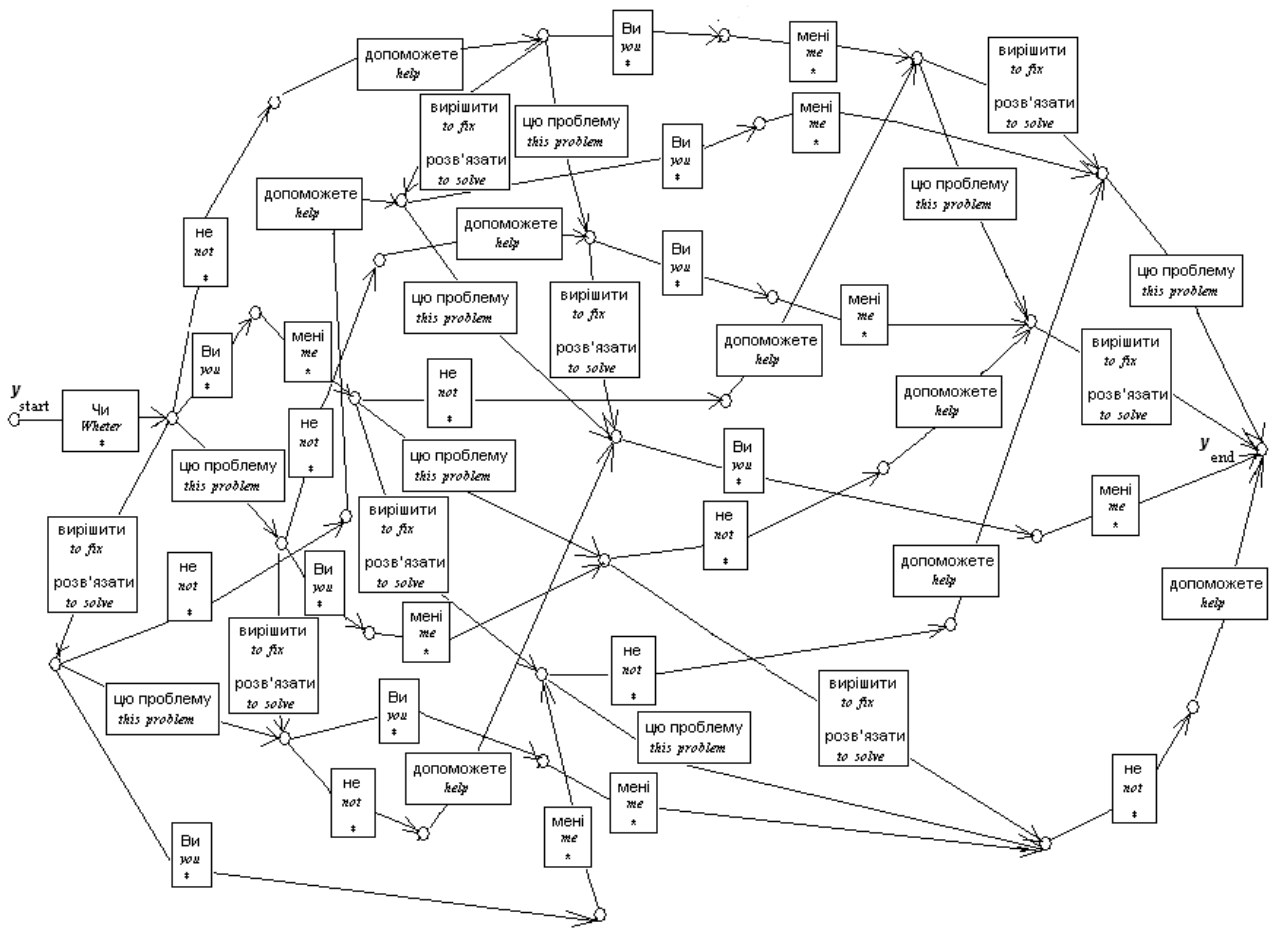

Figure 2. Oriented semantic network structure for the example sentence

4. Experimental results

As the experimental data an English-Ukrainian phrase book was used. It includes 3800 sentences, which we will call "basic". These phrases are divided into 15 subject areas and each of SA has its

MCs and STs. In experiments we examined one SA, in particular “Everyday phrases”. This SA contains 47 MC and 201 basic sentences, in average 5 basic sentences for each MC. To define the whole amount of sentences from the basic sentence we divided each basic sentence accordingly to described LISP-structures. Thus, for each basic sentence a machine readable LISP-structure was built.

For example for “asking for help” the following types of sentences were built:

((будь ласка | *) (допоможіть) (мені | *) ((вирішити | розв'язати) (цю проблему)))

((*please* | *) (*help*) (*me* | *) ((*to fix* | *to solve*) (*this problem*)))

((у цьому питанні) (мені | *) ((буде | *) (потрібна)) ([Ваша | *] (допомога)))

((*in this problem*) (*for me* | *) ((*will be* | *) (*needed*)) ([*your* | *] (*help*)))

The software was developed, which is able to generate the whole amount of sentences from one ST in accordance to specified inversions and substitutions of words and word combinations. As the result, from 201 ST we obtain 1045 phrases not including parameters like titles of cities. After including parameters we received 4337 phrases. Dictionary contains 290 words.

For proper Ukrainian into English translation we should extract the type of the pronounced sentence, respective meaning category and then choose the corresponding English phrase.

Recognition of phrases was carried out under both free and restricted grammars. Restricted grammar was set for each ST by means of LISP-structure.

For the experiment it was chosen 100 phrases from generated 4337. For these 100 we use the algorithm of phoneme recognition on conditions of free and restricted grammars [2, , 5]. While analysis of results we consider only those results which differ not more than on 2 verbal insertion/falling out or were not different in the contest. The results are in the table below:

Table 1. Results of phrases recognition.

Type of grammatics	% of proper sentences recognition to:			
	insertion/omission out			Type of contest
	0	1	2	
restricted	95	97	99	98
free	51	70	85	95

While using the restricted grammars we received in average 97% result of recognition and duration of the process was 30 minutes. As for free grammars the result was worst – in average 68% phrases identified properly though the algorithm in this case works much faster – 1,5 minutes.

Taking into account these results the demo software was developed to translate the phrase pronounced in Ukrainian into English. At the same time the order of words in the phrase could be variable. The English version of MC or sentence associates with the phrase in Ukrainian with the help of heuristic algorithm based on analysis of key words. The first sentence of this MC is assumed to be the result of translation.

5. Summary

The work includes questions concerning semantic interpretation of oral signal taking account the specific of Slavic language – relatively free word order and high inflexibility.

The method of setting a set of sentences with the same content using the building of ST with LISP-structures. The software was developed for building the oriented semantic networks according to ST and MC while recognition and creating the sentences while synthesis the result.

Experimental results have shown the high level of recognition and interpretation of continuous

speech in conditions of restricted grammars and hopeful results – in conditions of free grammars. Recognition in conditions of free grammars happens in real time. The fair assumption is that partially restriction of free grammars results in higher results even in real time.

The heuristic algorithm of comparison of recognition results with ST is proposed. Its using gives us good results also.

Based on experimental model the demo-model of oral translation from Ukrainian into English within the certain subject area is developed.

While generating sentences based on LISP-structures we receive also the sentences which are less typical for language. It is worth to be analyzed in future. The useful would be the creating of algorithm of automatic building of LISP-structure. Building of LISP-structures is quite bulky and requires a lot of handwork. So it should be automated.

One and the same text could be interrogative or narrative depending on intonation. So, in future, the intonation could also be analyzed and interpreted with the aim to use punctuation in results.

References

1. T. Vintsiuk, M. Sazhok. Multi-Level Multi-Decision Models in ASR. In Proc. Of 10th Int. Conf. "Speech and Computer", Patras, Greece, 2005, pp. 69-76.
2. T.K. Vintsyuk. Analysis, recognition and semantic interpretation of speech signals. – Kiev. Naukova dumka, 1987.
3. T.K. Vintsyuk. Language Syntaxis while recognition of continuous speech.. – Kiev. Cybernetics Institute, 1975.
4. Young S.J. et al., HTK Book, version 3.1, Cambridge University, 2002.
5. A. Lee, T. Kawahara and K. Shikano, "Julius – an open source real-time large vocabulary recognition engine." In Proc. European Conference on Speech Communication and Technology (EUROSPEECH), pp. 1691–1694, 2001.