Distinctive features for Ukrainian real-time speech recognition system

Mykola Sazhok, Valentyna Robeiko, Dmytro Fedoryn

Speech Science and Technology Department, IRTC, Kyiv, Ukraine

mykola@uasoiro.org.ua, valya.robeiko@gmail.com, dmytro.fedoryn@gmail.com

Abstract

This paper describes a real-time speech recognition system for Ukrainian designed basically for text dictation purpose targeting moderate computation requirements. The research is focused on features which are specific particularly for Ukrainian. Given arguments confirm the necessity to distinguish stressed and unstressed vowels in the phoneme alphabet. Lexical stress irregularity implies expert involvement for stress assignment. To automate this procedure we propose a data-driven stress prediction algorithm that represents words as sequences of substrings. The formulated criteria that validates a substring sequence is based on a set of words with manually pointed stresses and a large text corpus. The described search algorithm finds one or more sequences with the best criteria. As a Slavonic language Ukrainian is highly inflective and tolerates relatively free word order. These features motivates transition from word- to class-based statistical language model. According to our experimental research, 4-gram class-based LM occupies less space and has promising prospectives. We describe a speech-to-texe web-service where the proposed techniques are used as well as several tools developed to visualize HMMs, to predict word stress, and to manage cluster-based language modeling.

1. Introduction

Specific features of Slavonic languages are high inflectiveness and relatively free word order, which leads to rapid growth of the recognition vocabulary (8-10 times larger for same domain in English) and weakening of the language model prediction force. That is why the applicability of conventional methods and algorithms to Slavonic languages looks rather unpromising that is the reason of search for alternative to conventional recognition schemes, particularly considering word composition by the acoustic phoneme decoding output [1]. However, the potential of the recognition scheme having been developed for decades still remains uncovered [2].

The open question is limits of the vocabulary used in the speech-to-text system based on the conventional recognition scheme provided that the system shows real-time performance on a computational platform available for an ordinary user.

Therefore we aimed to build a real-time system that could be exploited on a contemporary personal computer for speech-to-text conversion like a dictation machine.

The system operating conditions must meet potential user's expectations. The recognition vocabulary should cover arbitrary speech with OOV < 1% and means to update the vocabulary must be provided. Acoustically, the system must be able to process speech of every adequate user. In advance prepared speech, read text and spontaneous utterances should be recognized on a similar level of accuracy. The system must provide an ability for the user to dictate in conditions of home and office inside and perhaps outside.

In previous work [3] we described a speech-to-text system that operated in real time with a 100k vocabulary tightly covering common and news domains (politics, economics, culture, education, sports, and weather). Nevertheless we must move towards a vocabulary for million words to reach the desired OOV for the arbitrary speech.

In this paper we explain assumptions concerning language distinctions on acoustical, phonetic and lexical levels, try to clear a prospective to attain the necessary vocabulary size, describe respective developed tools and discuss experimental results.

2. Speech-to-text system structure

The basic speech-to-text conversion system structure is shown in Fig. 1. The real-time component implements *Recognizer* itself that refers to *Data and Knowledge Base* developed off-line by means beside the illustrated components.

To create a speech recognition system we developed several data and program resources and used the toolkits available on Internet.

Real time component takes the *Input speech signal* from an available source (microphone, network or file system). *Voice activity detector* suggests beginnings of speech segments for *Pre-processor* that extracts acoustic features from. The system uses mel-frequency cepstral coefficients with subtracted mean and accomplished with energy and dynamic components (delta and delta-delta coefficients). *Decoder* compares an input segment with model signal hypotheses, being generated in accordance to acoustic and language models, using a conservative strategy of non-perspective hypotheses rejection [4]. The output, presented as a confusion network, is passed to *Decision Maker* that forms a *Recognition response* considering the history and performing necessary mappings to symbols and actions.

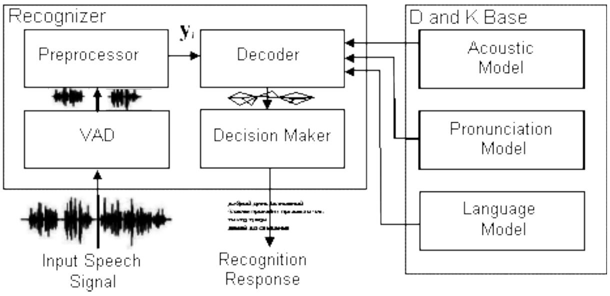


Figure 1. General structure for the basic speech-to-text system

Acoustic model is developed on subsets of the AKUEM speech corpus [5],[6]. The basic phoneme alphabet consists of 56 phonemes including stressed and unstressed versions for 6 vowels. The reason we distinguish them is discussed in the next chapter. Currently, HMMs built for context-independent phonemes contain from 8 to 32 Gaussians.

Pronunciation model provides *Decoder* with word pronunciation transcriptions formed off-line by Grapheme-to-phoneme module that implements a multilevel multi-decision symbol conversion technique based on describing the regularities of relation between orthographic and phonemic symbols [7]. An expert formulated about 40 local rules of grapheme-to-phoneme mapping partially modeling the individual speaker peculiarities and co-articulation and reduction of sounds in a speech flow. The rules are adjusted so that on average each word produces about 1.2 transcriptions. The same algorithm with another rules allows for converting numbers, abbreviations and symbolic characters to word sequences. The vocabulary for the entire system consists of a frequency dictionary extracted from the large text corpus and supplementary vocabularies covering speech corpus, social and local dialects, proper names, abbreviations etc. Taking a specified amount of top-frequent words from the system vocabulary a recognition vocabulary is formed.

Language model is created proceeding from the recognition vocabulary and a text corpus subset consisting of sentences containing below the specified portion of OOV words. The basic text corpus is derived from a hypertext data downloaded from several websites containing samples of news and publicity (60%), literature (8%), encyclopedic articles (24%), and legal and forensic domain (8%). To be noted that the data downloaded from news websites contains numerous user comments and reviews, which we consider as text samples of spontaneous speech. Text filter, used for text corpus processing, provides conversion of numbers and symbolic characters to relevant letters, removing improper text segments and paragraph repetitions. Total size of the basic text corpus is 2 GB that includes 17.5 million sentences that is a list of words containing above 275 million items and forming a vocabulary of more than two million words.

For the recognition vocabulary of 100 000 words, 88.5 million distinct 3-grams are detected in the subset of the basic text corpus after removing sentences containing more than 20% or at least three running unknown words. This sub-corpus is used for language modeling and referred as 250 M corpus. Consequently, we got OOV words occupy 2.5% of all words that is about twice less than in Ukrainian arbitrary text for the specified vocabulary size. To model spontaneous speech characteristics a class of transparent words is introduced to the recognition vocabulary. It contains non-lexical items like pause fillers and emotion and attitude expressions (laugh, applauds etc.).

Applying a language modeling tool [8] we have received a text file in ARPA format that occupies 5 GB reduced to 1.2 GB by a module of the decoder tool [4].

The real-time modules are used to build a basic speech-to-text conversion system for experimental research and trial operation. Graphical user interface integrated with the basic system allows for demonstrating continuous speech recognition for wide domain in real time, using a contemporary notebook [3].

Now we are going to analyze features that are specific particularly for Ukrainian, to explain assumptions concerning language distinctions on acoustical, phonetic and lexical levels, and to introduce extensions to the basic speech-to-text system.

3. Lexical stress analysis

In many languages, certain syllables in words are phonetically more prominent in terms of duration, pitch, and loudness. This phenomenon is referred to as lexical stress. Do we need to introduce both stressed and unstressed vowels to the phoneme alphabet?

Answering positively to this question we rely on phonetic, lexical and acoustical facts for Ukrainian. Stressed vowels normally acts as phonemes changing word grammatical function and meaning that we observe in about 10% of words in arbitrary texts.

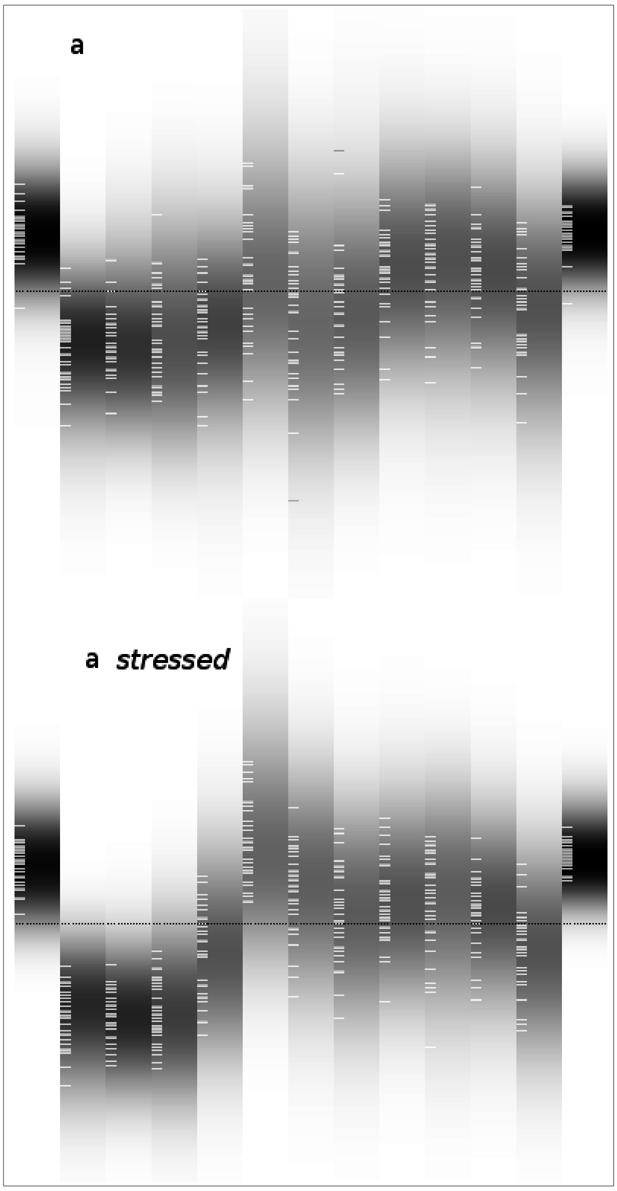


Figure 2. HMM visualization for Ukrainian monophone **a** in unstressed (above) and stressed (below) positions

To explore the acoustical side of the problem we trained stressed and unstressed vowels as if they are different phonemes and inspected dissimilarities particularly by means of the HMM visualization tool [9]. Following Figure 2 we can see the difference between unstressed and stressed phonemes

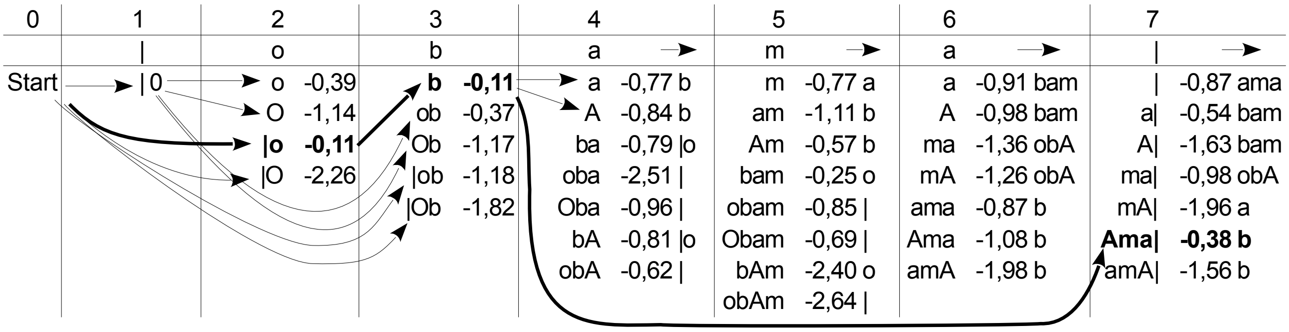


Figure 3. Stress prediction for an out-of-vocabulary word “obama”

a. More phonemes are available for examining on the tool’s webpage.

In Ukrainian, stress position is irregular and it can be changed even within forms of the same word. Anyway, it is not acceptable to point stresses manually for the entire lexicon. Therefore, we propose a word stress prediction procedure based on the known vocabulary and a text corpus.

We consider all possible segmentations S for a word with unknown stress. The *i*-th segmentation of S

$$S_i = (q_{i,1}, q_{i,2}, \dots, q_{i,j}, \dots, q_{i,L_i}) \qquad (1)$$

has length of L_i. Here $q_{i,j}$ is a *j*-th item (a character or a phoneme) within the *i*-th segment of S. Now we introduce a vector θ_L that indicates the stress level (e.g. 0, 1, 2) for each o f L items. Thus, we can estimate a probability of stress position given the segment S_i :

$$P(\theta_{L_i} \mid S_i) \approx \frac{c(S_i, \theta_{L_i})}{c(S_i)} \qquad (2)$$

where $c(S_i, \theta_{L_i})$ is count of segments S_i with stress position defined by a stress indication vector θ_{L_i} and $c(S_i)$ is the number of S_i total occurrence. All counts are taken from the text corpus but the words are not included in stress vocabulary.

Finally, we search through all valid segmentations S and stress positions θ^S that satisfy the expression:

$$\operatorname*{argmax}_{S,\theta^S} \prod_{S_i,\theta_{L_i}} P(\theta_{L_i} \mid S_i) \; . \qquad (3)$$

We constructed a dynamic programming graph where finding the shortest trajectory is equivalent to the search (3). Memorizing N prospective arrows in nodes of the graph we can extract N-best word stress positions supplemented with the probability estimation.

We estimated stress prediction model parameters on 250 M text corpus. Special word boundary symbol was included. More than 60 000 character segments detected for length one to four. In Figure 3 an example of one-best stress prediction is shown for a proper name “Obama” missing from the basic Ukrainian vocabulary. The word is represented as concatenation of all valid character segments where the largest segment length is limited to four. Each input character introduces a set of valid segments. Potentially optimal arcs are either shown or coded with the name of a previous node. Partial criteria are log probability based. The optimal path |o-b-Ama|, respective nodes and criteria are bold. Pay attention that we avoid connecting segments with potentially optimal criteria obA-mA since both of them are stressed.

Stress error rate estimation in not as obvious procedure, since in specific cases it is unclear what a mistake is, e.g. the stress is predicted in misspelled words but if the prediction is mistaken why should we be as strict? Anyway, preliminary experiments exposed error level between 5 and 10% relatively to the vocabulary size.

4. Class-based LM development

As a Slavonic language, Ukrainian is highly inflective, the number of word forms per dictionary entry accedes 12 that is about 6 times more than for English. Therefore, to build an adequate language model a 6 time larger vocabulary is required. Moreover, relatively free word order is normative that leads to perplexity and data sparsity growth. Analysis of these features motivates a transition from word- to class-based statistical language model that operates with transition probability and membership probability [10].

Word clustering procedure tries to minimize the perplexity improvement criterion

$$F_G = \sum_{g,h \in G} C(g,h) \log C(g,h) - 2 \sum_{g \in G} C(g) \log C(g) \qquad (4)$$

where (g, h) means a class g follows a class h from the set of equivalence classes $\boldsymbol{G}$ and function $C(\cdot)$ counts its argument occurrence in the training corpus. An exchange algorithm described in [10] implies iterations in which each word is tested for a better class and consequently moved there. While implementing the algorithm we came to an alternative formulation of criteria computation refinement [11].

The clustering results have been analyzed proceeding from their relevance to linguistic categories. Firstly automatically obtained classes for Ukrainian in general correspond to syntactic, semantic and phonetic features.

Most word classes have an obvious syntactic interpretation, such as nouns in a genitive form, or plural adjectives. Table 1 shows several word classes that have been obtained by bigram clustering on the 250 M corpus for 1000 word classes. The words in each word class are listed in descending word unigram count order and the most frequent word is emphasized. We present three classes completely and first 7 words for the last class. Often, there is some semantic meaning like in the last class containing verbs of communication (for third person in present and past tenses). Two first classes show that misspelled but still frequent words may join to the class containing a correct version of the word.

In Ukrainian, words may have different forms in dependence of phonetic context. For instance, the conjunction and has three forms normally used between consonants, between vowels and in other cases. All these forms were automatically assigned to different classes.

Table 1. Bigram clustering examples for 1000 classes

Words of cluster with meaning	Frequency
багато / many, much	134590
чимало / plenty	24482
безліч / a lot of	7696
немало / quite a lot of	2191
якнайбільше / as many	760
багацько / lots of	255
богато (*misspelled* багато)	123
які / that, which (plural)	590681
котрі / that, which (plural)	24499
яки (*misspelled* які)	465
де / where	246376
куди / to where	31966
звідки / where from	15373
звідкіль / where from (colloquial)	120
заявив / [he] stated	163547
вважає / [he, she] supposes	99803
повідомив / [he] informed	80043
заявила / [she] stated	32795
заявляє / [he, she] states	31965
розповів / [he] told	30504
говорить / [he, she] speaks	29756

5. Experiments

We considered two experimental speech corpus sets. Test Set 1 contains 49 preselected transmissions and Set 2 contains 78 randomly selected ones. Table 2 shows that both sets has comparable lengths and Set 1 is mostly oriented on forensic domain. For each set we estimated acoustic parameters on AKUEM speech files excluding the respective test set.

Table 2. Test sample summary

Set ID	Length	Forensic show	Judge speech	News	Show	Press-conference
1	11.4h	69.4%	11.1%	8.4%	8.2%	2.9%
2	12.6h	32.5%	-	29.8%	36.8%	0.9%

Language model were build on 250 M text corpus. No text samples from AKUEM transcriptions were taken. Most frequent words formed 100k and 200k vocabularies. Only fist 100k words were automatically assigned to classes. Less frequent words were assumed unknown.

As it follows from Table 3, stressed vowel introduction leads to convincing error reduction. The class-based LM anyway shows its potential despite certain falling behind the word-based LM.

Table 3. Experiment results

Set	Stresses	Classes	LM order	Vocabulary size / %OOV	%WER
1	-	-	3	100k / 5.27	33.6
1	+	-	3	100k / 5.27	32.1
1	+	1000	3	200k / 3.79	34.1
1	+	1000	4	200k / 3.79	33.8
2	-	-	3	100k / 5.61	38.0
2	+	-	3	100k / 5.61	36.3
2	+	1000	3	200k / 4.15	38.7
2	+	1000	4	200k / 4.15	38.5

6. Speech-to-text system development and trial operation

In previous works we described the dictation machine that is a real time speech-to-text conversion system providing basic interactions with a user [3]. However, a huge variety of other solutions for speech recognition technology might be topical as well.

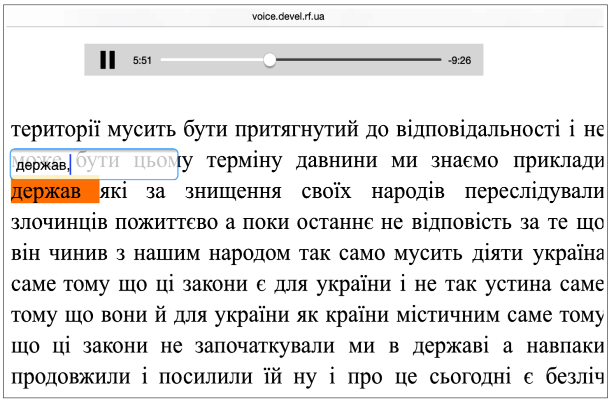
Depending on the place where speech-to-text conversion takes place the recognition software and systems can be divided into isolated (client-side), client-server (server-side) and hybrid systems. In isolated case all transformations occur directly on the client device. In client-server systems a client device is used just to record speech signal, transfer it over the network to the server for further processing and to receive responses from the server. Hybrid systems combine the functionality of isolated and client-server systems: when a network access is available the recognition server is used and an isolated system runs otherwise. An example of the isolated system is *CeedVocal* speech recognition engine [12]; an example of client-server system is well-known *Siri* [13] and speech input systems developed by *Google* [14]; and *VoCon Hybrid* is a hybrid system [15].

Each approach has its advantages and disadvantages. Isolated system performance is limited by RAM and CPU speed of client systems, which in turn restricts for the size of the vocabulary. Client-server technology does not have these limitations but requires for its operation a permanent connection to the global network. Hybrid technology is essentially the implementation of previous two technologies in one system, therefore its development requires more time and resources than implementation of each technology separately.

The client/server model was chosen for fast speech recognition system. The REST interface is used for data exchange between client and server. It means, that remote procedure call is a common HTTP-request (POST or GET), and the necessary data is sent as query parameters. The software was written in PHP (data acquiring from clients and web-content generation) and C++ (Decoder).

We developed a Ukrainian speech recognition service based on client/server approach [16]. Users can upload their own files to the server using special command line programs (such as *cURL* [17]), or through the web interface. At the web interface there is the possibility to load media files directly from *YouTube* service [18]. As a result the client receives a webpage with mediaplayer and recognized text synchronized with audio signal. It means, that the fragment of recognized text, that corresponds to what was said at the time, is synchronously highlighted when playing a media file. The system vocabulary includes 200 000 words. Recognition is twice faster than real-time. The system uses power of the 4-core *Intel Xeon* processor, so simultaneous execution of four different tasks is possible without loss of recognition time performance.

The experimental service on web allows for converting media data to text and make word corrections while listening to the respective speech segment. In Fig. 3 we can see a fragment of speech-to-text conversion results for the record of a politician press-conference. Total duration of the record is 917 s, WER=25.8%. For majority of erroneously recognized words the user needs to correct just one symbol. Only single word at once might be corrected or two words might be merged.



території мусить бути притягнутий до відповідальності і не
держав, бути цьому терміну давнини ми знаємо приклади
держав які за знищення своїх народів переслідували
злочинців пожиттєво а поки останнє не відповість за те що
він чинив з нашим народом так само мусить діяти україна
саме тому що ці закони є для україни і не так устина саме
тому що вони й для україни як країни містичним саме тому
що ці закони не започаткували ми в державі а навпаки
продовжили і посилили їй ну і про це сьогодні є безліч

Fig. 3. *The web-based speech-to-text conversion tool allows for editing words synchronously with sound.*

7. Conclusions

The described real-time system for Ukrainian speech-to-text conversion demonstrates a potential of focusing on language distinctive features, which makes feasible to attain vocabulary size necessary to reduce OOV below 1% and to introduce punctuation and character case dependency.

The proposed stress prediction procedure allows for assigning most hypothetically possible one or more lexical stresses in unknown words. However, stress disambiguation even for known words is necessary for further introduction of the semantical level.

Distance to closest alien classes should give a clue to predicting homographs and consequent semantic word decomposition that may lead to more homogeneous classes.

The implemented word clustering is an efficient way to reduce LM space and to introduce significant to the user's domain new words to the vocabulary.

For dictating purpose, human-machine interaction is crucial. The system has to suggest recognized utterance refinement based on multi-decision recognition response; moreover, accepted refinements must update the recognition response model. Besides assigning a new word to the unknown word category, we plan to implement updating the class language model by mapping new words to classes and recomputing class membership probabilities.

For text processing, further work on number- and symbol-to-grapheme conversion is topical in sense of predicting their correct agreement for the observed context.

The development of the presented system is on early stage. In near future several improvements will be completed, which will increase accuracy and extend the scope of usage.

8. Acknowledgements

Authors express gratitude to Special Recording Systems, Ltd for provided computing and network resources. Authors are also thankful to letvox.com for kindly provided server resources.

References

[1] Taras Vintsiuk, Mykola Sazhok. Multi-Level Multi-Decision Models for ASR. In Proc. SpeCom'2005, Patras, 2005, pp.69-76.

[2] M. Gales and S. Young. "The Application of Hidden Markov Models in Speech Recognition". Foundations and Trends in Signal Processing, 2007, 1(3), pp. 195-304.

[3] V. Robeiko, M. Sazhok. Real-time spontaneous Ukrainian speech recognition system based on word acoustic composite models. In Proc. UkrObraz'2012, Kyiv, 2012, pp. 77-81.

[4] A. Lee, T. Kawahara. Recent Development of Open-Source Speech Recognition Engine Julius. APSIPA ASC, 2009, pp. 131-137.

[5] Young S.J. et al., The HTK Book Version 3.4, Cambridge University, 2006.

[6] Valeriy Pylypenko, Valentyna Robeiko, Mykola Sazhok, Nina Vasylieva, Oleksandr Radoutsky. Ukrainian Broadcast Speech Corpus Development // Specom'2011, Kazan, 2011, pp. 244-247.

[7] V. Robeiko, M. Sazhok. Bidirectional Text-To-Pronunciation Conversion with Word Stress Prediction for Ukrainian. In Proc. UkrObraz'2012, Kyiv, 2012, pp. 43-46.

[8] Bo-June (Paul) Hsu and James Glass. Iterative Language Model Estimation: Efficient Data Structure & Algorithms. In Proc. Interspeech, 2008.

[9] www.cybermova.com/speech/visual-hmm.htm

[10] S. Martin, J. Liermann, and H. Ney, "Algorithms for bigram and trigram word clustering," in Proceedings of Eurospeech, vol. 2, pp. 1253–1256, Madrid, 1995.

[11] M.Sazhok, V. Robeiko. Language Model Comparison for Ukrainian Real-Time Speech Recognition System. M. Železný et al. (Eds.): SPECOM 2013, LNAI 8113, pp. 211–218, 2013.

[12] www.creaceed.com/ceedvocal/about/

[13] www.apple.com/ios/siri/

[14] www.google.com/intl/en/chrome/demos/speech.html

[15] www.nuance.com/for-business/by-product/automotive-products-services/vocon-hybrid/

[16] www.cybermova.com/technology/synchrophone.html

[17] curl.haxx.se/

[18] www.youtube.com/