Rapid Prototyping of a Croatian Large Vocabulary Continuous Speech Recognition System

Dario Bajo, Danijel Turković, and Šandor Dembitz
University of Zagreb, Faculty of Electrical Engineering and Computing (FER)
e-mail: {dario.bajo, danijel.turkovic, sandor.dembitz}@fer.hr

Abstract—**The Croatian language, like many minority languages used by less than 0.1% of the world population, is in need of mature automatic speech recognition (ASR) systems for applications such as transcription of speech recordings, voice control, an aid to impaired people, etc. This paper describes a short-term research and development project aimed to produce an applicable Croatian large vocabulary continuous speech recognition system from scratch. The open-source CMU Sphinx toolkit was our platform choice. For the purpose of acoustic model training, we made a speech training set of several hundred utterances, containing words carefully chosen according to their phonetic properties. Language models were derived from the Croatian large-scale n-gram system, which ensures the system's applicability. During the project, we succeeded in developing an ASR system able to recognize freely chosen utterances composed of 15,000 most frequently used Croatian words reasonably well.**

Keywords-automatic speech recognition; continuous speech; large-scale n-gram model; large vocabulary.

I. INTRODUCTION

The first research and development attempting to produce an applicable ASR system for the Croatian language was done at Carnegie Mellon University (CMU), Pittsburgh, PA, USA, motivated by the needs of the USA Army personnel located in the Balkans in that time [1]. Since the USA Army's priorities changed drastically after September 11, 2001, the project ended without delivering intended field system. After that, some attempts were made in Croatia too, but they did not result in publicly available ASR systems yet.

Non-existence of Croatian large vocabulary continuous speech recognition (LVCSR) systems was the main impulse for our research. In this paper, we examine the current state of Croatian ASR systems, shortly explain the most important theoretical concepts and techniques used in ASR and present our own implementation of Croatian ASR using CMU Sphinx, a state-of-the-art speech recognition toolkit created by one of the leading language technology laboratories in the world [2]. Our system is designed to recognize 15,000 most frequent Croatian words, or at least 75% of Croatian word usage, using the Sphinx4 speech recognition library.

An overview of the process of building such a system will be given with detailed examination of the parts which were most interesting during the research.

A. *Related work*

In [3], a Croatian ASR system is realized using the Sphinx3 speech recognition library. It was tested on many speakers and utterances. The vocabulary consisted of only 40 words, which resulted in very effective recognition. The English phonetic transcription was used to build a Croatian pronouncing dictionary in order to provide a recognition system exclusively for the Croatian language.

In [4], an acoustic model for Croatian LVCSR system, with vocabulary size of 14,551 words, was created using Croatian speech database of weather forecasts (VEPRAD), as well as read tales and spontaneous speech. A statistical bigram language model was used, which was derived from the training utterances. Their system was based upon the HTK speech recognition toolkit [5], which differs in many details from CMU Sphinx, and they report achieving word error rate (WER) below 5% for a test set which consisted only of weather related utterances from their training set.

In [6], Slovak ASR system is implemented using the Sphinx4 library for digits and application words recognition in GSM networks. The MOBILDAT-SK database for the Slovak language was used. The WER results were fairly low (below 10%). The context-dependent system gave slightly better results.

In [7], a task-oriented continuous speech recognition system for the Polish language is implemented as a voice interface for a computer game *Rally Navigator*, using the Sphinx4 library. They have managed to achieve sentence recognition accuracy of 97.6%.

Very successful ASR systems exist for English, even with unconstrained vocabularies. The Whisper system described in [8] is given as an example.

B. *Organization of the paper*

In Section II, some theoretical concepts are given, which are needed to understand the basics of acoustic and language modeling.

Section III presents the methods used to build our ASR system. First, in subsection A, the pronouncing dictionary creation is mentioned. In subsection B, all the required steps one would have to take to build an acoustic model in CMU Sphinx are described. In subsection C, a detailed overview of the development of our acoustic model is given. Subsection D demonstrates our approach to language modeling. In subsection E, we explain our ideas for further improvements of our language model. The development described in Section III lasted from October 2012 until March 2013.

In Section IV, we discuss our speech recognition results and present independent WER measurements that confirm them.

ISBN: 978-1-61208-310-0

In Section V, we demonstrate further developments of large-scale language models for Croatian ASR purposes and discuss the feasibility of extending the vocabulary size in order to achieve larger word usage coverage.

Finally, in Section VI, we conclude the paper with a summary of our results and a thought about the possible use of our research in future projects aiming to create Croatian LVCSR systems.

II. Theoretical Foundations

In order to understand how a speech recognition system works, one has to know the underlying concepts behind modeling two key parts of every ASR system: the acoustic model and the language model.

A. *Acoustic Modeling*

One of the main problems of this method is the adjustment of speech signal for processing and analysis of basic speech units - phonemes [8]. For effective phoneme detection, the main information carriers have to be extracted from a speech signal (features). Such carriers are mel-frequency cepstral coefficients [9]. For successful further processing, it is crucial for samples to be accurate (recognizable in different contexts), trainable (feasible parameter estimation), and generalizable (new words composition) [8].

A word, as a basic processing form and meaningful speech unit, is certainly one of the most important information carriers, but its recognition is very difficult since a language contains many words, including different word forms in highly inflected languages like Croatian. In the testing phase, it can lead to many obstacles (non-generalizable samples) and the results of the phonetic analysis may be inaccurate. Syllables, however, are inadequate as a training set, especially for the Indo-European languages [10].

A compromise is achieved by suing a specific array of three phonemes - triphone, which describes a clear pronunciation of the central phoneme - allophone (neighboring dependency). One of the biggest allophone and phoneme advantages in relation to the other units is parameter sharing. This significantly reduces computing time for parameter estimation, i.e., the parameters for allophones and phonemes can easily be estimated from acoustic parameters of a known training set.

Allophones may also differ by intonation (position in sentence) and degree of stress (vowels at higher dose of stress last longer, have higher pitch, and are more intense). Many similar phonemes (labials, velars) are grouped into corresponding classes (clusters) consisting of senones.

A senone is a specific phonetic subunit which describes the type of allophone. Their amount depends on the learning corpus, which highly influenced the construction of our final training set described in III.C. Decision trees contain senones and searching is enabled by many binary conditions as internal nodes.

Hidden Markov models (HMMs) are used for modeling segmented allophones from a language learning corpus [11]. HMM consists of states, transitions, and distributions. It represents the realization of a complex discrete finite state machine in which every next state depends only on the previous state. The state probability is computed as a product of initial state probabilities and transition conditional probabilities. As mentioned in [8], there are two characteristic assumptions employed in HMM studies - Markov's and event independence.

The HMM modeling and processing is decomposed into three problems: evaluation, decoding, and training [8]. The evaluation problem is subdued to the calculation of state posterior probability using Bayes' formula [12] and state output probabilities.

The decoding process results in the most probable set of states (deterministic states), where HMM becomes an ordinary Markov model. The most probable state sequence path and decoding is implemented by the Viterbi algorithm [13] and applied to the process.

The learning process is conducted several times during the construction of the acoustic model. The main goal is to estimate the model parameters using the Baum-Welch (forward-backward) algorithm, while HMMs can be implemented as continuous, semi-continuous or discrete [14].

Acoustic modeling is used to compute acoustic parameters by using loaded utterances and phonetic rules. Previously obtained feature vectors, along with the speech phonetic transcriptions and monophone labels, are used to train the parameters of newly created monophone HMMs. After monophone HMM is formed, the automatic segmentation process is applied. Monophones are aligned to the recorded speech sequence, and then trained with the Baum-Welch algorithm by incrementing Gaussian density mixtures [4][8].

After the training process, the triphone structures are built using estimated parameters and generated triphone labels. The senones [4][15] are classified using decision trees. After the state reduction, triphone HMMs are ready for merging into bigger units such as subwords and words. The phonetic dictionary and HMM-formatted phonetic transcriptions are employed to achieve merging.

Furthermore, the Baum-Welch training algorithm is applied once more including some slight modifications (insertion, replacement or deletion of allophones-triphones) in order to achieve correct utterance transcription. Triphones obtained by above-mentioned method represent acoustic modeling output and together with the n-gram language model form a system for speech recognition.

B. *Language Modeling*

The concept of language modeling is closely associated with word searching space reduction during the construction of sentences. The reduction degree depends on learning corpus size, number of phonetic transcriptions, dictionary size, and degree of the implemented grammatical model. In this project, we used statistical language modeling based on the Croatian large-scale n-gram system [16].

Value n denotes the number of words in a particular structure, i.e., for n=3, the likelihood of the 3rd word will be computed on a basis of two words appearing before, according to the Bayes' formula [12].

In the case of word construction beyond the learning set, n-gram in utterance is marked with a minimal amount of probability. This method is also known as n-gram smoothing. It is executed by adjusting the maximum likelihood estimation [17] probabilities to obtain higher robustness.

III. Applied Procedure

The CMU Sphinx toolkit is based upon three main components: pronouncing dictionary, acoustic model, and language model. Pronouncing dictionary maps written word form into its pronunciation according to the predefined set of phonemes.

Acoustic model needs to be trained from pairs of spoken utterances and their transcriptions. After the previous step has been completed, the trained acoustic model with already prepared language model can be used in speech recognition of test utterances or live continuous speech.

A. Building the Pronouncing Dictionary

Because of Croatian phonemic orthography, when only words strictly obeying the orthography are considered, as this was in our case, the pronouncing dictionary creation is a straight mapping of written words into CMU Sphinx dictionary format.

B. Building the Acoustic Model

Necessary files needed to train an acoustic model are: phonetic dictionary, phoneme list, filler list, list of training audio files' IDs, and transcriptions of training audio files.

Phonetic dictionary must consist of all the words that occur in the training utterances with corresponding phonemes from which their spoken analogue is made of; phoneme list contains all the phonemes that occur in the spoken utterances; filler list is a file which consists of all the non-spoken sounds, such as the breathing sound, pause, etc.

List of training audio files' IDs consists of file names of all the training data that were used during the procedure of acoustic model training, while the transcriptions of training audio files contain all the utterances in their written form that correspond to the IDs of recordings mentioned earlier.

All the audio recordings were taken in MS WAV format with the sample rate of 16 kHz, 16 bit, mono. The lengths of recordings range from just a couple of seconds up to 30 seconds.

C. Developing the Acoustic Model

At the beginning of this project, we constrained ourselves to the subset of words which only covered letters of Croatian alphabet and digits from zero to nine. That system was tested and proved our suspicions that in this case the phoneme-based recognition system could not function properly because it did not have enough context to rely on.

The only result worth mentioning is the recognition of digits because those words consist of more phonemes and the system coped with them with ease.

After that stage, we moved to the domain of continuously spoken words by using a small training set of 270 short utterances, composed of 1,010 words. Initially, we constrained our recognition system to recognize just the words which have been already seen through the acoustic training process, although not necessarily in the same context as before.

After initial success, we decided to make a system able to recognize 15,000 most frequent words in the Croatian language, regardless of the amount of different recorded words in the training utterances.

In our new training set, there was a total of 657 utterances, built up of 4,145 different words, which were recorded by 15 non-professional speakers, 4 female and 11 male students. They produced 16-hour-long speech database for acoustic modeling. The utterance construction was governed by the idea of covering as many Croatian phoneme combinations and acoustic transitions as possible within a small sentence sample.

The recordings were not made in acoustically perfect conditions. On the contrary, the recordings were made in an environment which was likely to have noise and (at the time) we thought that it could ultimately make our ASR system more robust in a real-world situation.

After all prerequisite files are in place, the acoustic model training can begin. Important thing to notice is that many parameters of the procedure of acoustic model training can be changed from predefined values, but the parameters with the highest impact on the accuracy of the recognition itself are the number of Gaussian mixtures and the number of senones. The influence of these choices on the recognition accuracy is presented in Section IV.

D. Generating the Language Model

After the acoustic model has been trained, in order to use the recognition system, the language model must be generated. There are certainly many ways to do so, from web crawling to manually creating a set of possible word combinations which may occur in a speech that will be recognized.

Our approach was to use already existing corpus of unigrams, bigrams, and trigrams, which was obtained through Hascheck, Croatian academic spellchecking web service [16][18]. Out of all the words found in Hascheck's large-scale n-gram database, we selected only the 15,000 most frequent words that cover over 75% of the Croatian word usage and added 396 words which appear in our training set because of phonetic reasons.

The complete vocabulary was used to extract all the bigrams and trigrams in which only those words occur. Because of immense initial numbers, only n-grams with frequencies ≥ 10 were selected (10^+ n-grams in Table I).

TABLE I. Basis for language modeling.

all unigrams	all bigrams	all trigrams	10^+ bigrams	10^+ trigrams
15,396	≈14e+6	≈64e+6	2,955,551	5,214,340

After the selection was made, only the trigrams (without their frequencies) have been passed on to the CMU language modeling toolkit (CMU LMTK), along with the transcriptions of all the recorded utterances.

This approach yielded good results, but since the frequencies of trigrams from Hascheck's database were completely ignored and all the trigrams were treated as equally likely, couple of problems emerged, one of which was a problem with the recognition of the starting word in testing utterances, being particularly troublesome for our system. This is not surprising since at the beginning of an utterance there is no context to rely on and every word had equal probability of being recognized as the correctly spoken starting word.

E. Improving the Language Model

Because of the problem mentioned above, it was necessary to address this problem by pondering trigrams in order to preserve some information about the frequency of each individual trigram. After initial testing, it was discovered that additional pondering could be useful.

Therefore, we also decided to ponder unigrams, hoping to improve our recognition results achieved so far. The ponderings of trigrams and subsequently unigrams were made roughly on logarithm scale according to their original frequencies. This choice, motivated by the data minimization needs, proved to yield much better results, both in recognition accuracy and utterance decoding response time.

IV. RESULTS

The speech database was divided into the training set (roughly 80% of the total number of recorded utterances) and the testing set (the rest of the recordings) by random partitioning of each speaker's recordings into two groups (training and testing subset) in the same ratio.

Four systems with different language models were tested: trigram model (not pondered) extracted from the 15,000 most frequent words in the Croatian language, trigram model extracted from recorded utterances, pondered unigram and trigram model extracted from the 15,000 most frequent words, and pondered unigram and trigram model extracted from the 15,000 most frequent words merged with the trigram model found in recorded utterances. Pondered n-grams were repeated in sentence style as many times as needed, i.e., according to their pondered frequencies, in order to satisfy CMU LMTK requirements.

Unsurprisingly, the best language model for the testing set described above was the one consisting only of the trigrams which were extracted from the recorded utterances. This model outperformed all the other models, which is expected because the model was biased by the word combinations found in recorded utterances, but its vocabulary would be too specific for arbitrary speech.

Recognition results are presented at Fig. 1 by corresponding values of word error rate (WER), the ratio between the number of inserted, deleted, and substituted words in test utterances, and sentence error rate (SER), the ratio of utterances in which at least one word was incorrectly recognized.

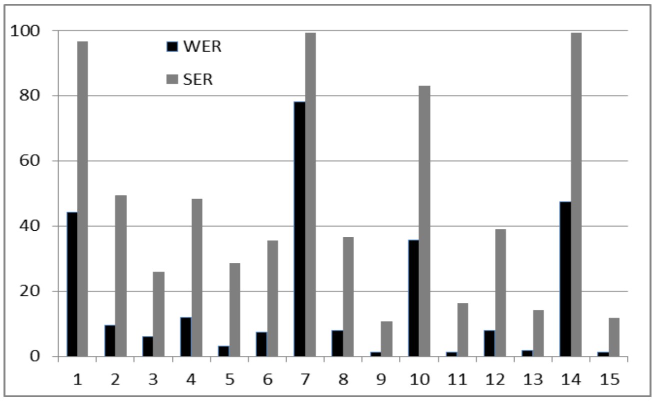

Figure 1. Speaker-dependent recognition results for the first testing set.

Presented results are achieved using large values of parameters for word combinations search and the computation needed to produce these results cannot be done in real-time (the speed of decoding an audio file was about 3xRT). We tried to use smaller values for those parameters and managed to get a system which can operate in real-time (the speed of decoding an audio file was roughly 1xRT), but the achieved WER results were fairly poor (about 65%).

The acoustic model generation for our final acoustic model lasted about 10 hours, while it took roughly an hour to create our biggest language model.

The whole computation/decoding were done in a virtual machine running Ubuntu 10.04 using VMware Workstation on a laptop with 4 GB RAM and 2.4 GHz processor.

Considering all 15 speakers in the first testing set, we obtained an average WER equal to 23.8%, and an average SER equal to 45.9%. The average results are heavily influenced by the utterances produced by speakers 1, 7, 10, and 14, whose recordings were done in extremely noisy conditions. Without them, the average WER would be 4.5%, and the average SER would be 28.6%. These recognition results are comparable to those presented in [4]. The measurements were performed with the number of Gaussian mixtures set to 8 and the number of senones set to 150 (default values).

Among other language models the last one, composed of pondered unigrams and trigrams containing 15,000 most frequent Croatian words and the utterances from the training set, has demonstrated to be the most accurate. Logarithmic pondering of n-grams and their conversion in sentences according to the pondered frequencies proved that our approach to language modeling through Hascheck's n-gram database was an appropriate choice for rapid LVCSR system prototyping.

Further testing was done by changing the number of Gaussian mixtures as well as the number of senones. The second test set was now built of 131 utterances freely composed of words from the dictionary, pronounced by a speaker whose voice is represented in the training set. The recognition results are presented in Table II.

Since the number of Gaussian mixtures can only be a power of 2, we tested powers of 2 between 8 and 64, while the number of senones remained at 150, except in one test scenario, when the tested value was 3000 (which was estimated by the CMU Sphinx toolkit as the best choice for our amount of training data).

The best result achieved for the number of Gaussian mixtures set to 8 and the number of senones set to 3000 is a consequence of the voice known to the system. For unknown voices, the best combination, according to our experience, is 32 Gaussian mixtures and 150 senones.

TABLE II. IMPACT OF THE KEY PARAMETERS ON THE RESULTS.

Combinations of the key parameters	WER	SER
8 Gaussian mixtures, 150 senones	28.4%	68.7%
16 Gaussian mixtures, 150 senones	24.4%	63.4%
32 Gaussian mixtures, 150 senones	21.8%	59.5%
64 Gaussian mixtures, 150 senones	19.7%	56.5%
8 Gaussian mixtures, 3000 senones	15.0%	49.6%

These results were confirmed by independent measurements performed according to the CMU Sphinx performance regression tests adapted to the Croatian language [19]. The main results are the following:

- Testing with known utterances from the training set gave WER of 7.83%;
- Testing with known utterances spoken by a speaker who was not in the training set gave WER of 10.82%;
- Testing with unknown utterances composed of words that are covered by our phonetic dictionary and spoken by a speaker who was not in the training set gave WER of 24.81%.

Unstressed monosyllabic words demonstrated to be the most problematic for correct recognition. For example, the Croatian number 5, whose pronunciation is “pet”, very similar to the pronunciation of the identically written English word, was almost regularly misdecoded.

V. FURTHER DEVELOPMENTS

As one direction of possible improvements, we tried to test the limits of our LVCSR system by increasing the number of words our system can recognize, and check the feasibility of using the larger system for speaker-independent speech recognition.

A. *Enlarging the Pronouncing Dictionary*

From Hascheck’s unigram database we took 130,160 most frequent Croatian words, which cover 95% of the Croatian word usage, as a basis for developing a large Croatian pronouncing dictionary. Those words were divided into three groups:

- Croatian non-name words;
- English words (mostly international names like Alexander, Mexico, Yamaha, etc.) contained in the CMU Sphinx pronouncing dictionary;
- All the other words.

Since the file with Croatian common (non-name) words contained 102,636 items, going through these words manually and writing them in the CMU Sphinx dictionary format would require a lot of time. Therefore, we made a short script in the Python programming language, which reads word by word from the input file containing Croatian words and writes those words in the CMU Sphinx dictionary format to the output file. Because of Croatian phonemic orthography, this was an easy task.

Generating Croatian pronouncing dictionary for English words (12,635 such words were found) proved to be a bit trickier. The English language contains many phonemes (“ah”, “iy”, “th”, and the like) which do not exist in the Croatian language. Therefore, we had to write a program which converts them into their Croatian counterparts, in a manner in which they would be pronounced by native Croatian speakers. The program was tuned by testing how the English words, being converted into Croatian phonetic system, are pronounced by HascheckVoice, Croatian academic speech synthesizer [20]. The final version of the program was applied to the English words found in the large Croatian dictionary, and this resolved the problem of their phonetic encoding.

Among other words (14,889 words in total) dominate name entities with South Slavic origin, which obey the same orthography as common (non-name) words. They were extracted and converted into the CMU Sphinx dictionary format in the same manner as common Croatian words. The remainder (3,102 words in total) had to be encoded manually. This was done in a few days.

All the dictionaries discussed in the next subsection are subsets of the large Croatian pronouncing dictionary with 130,160 entries.

B. *Generating Bigger Language Model*

After the initial success with 15,000 words covering 75% of the Croatian word usage, our goal was to develop a language model for 95% of the Croatian word usage. The intention was to repeat the steps used for generating the language model based upon 15,000 words on the new vocabulary size of 130,160 words, in order to produce a new language model for speech recognition purposes.

Since the size of n-gram files generated from 130,160 words was too big to handle, only n-grams with frequencies ≥ 10 were selected, which resulted in 4,158,737 bigrams and 15,686,105 trigrams. Selected trigrams were given as an input to the CMU LMTK.

Here is the point where the problems started. Generating the .arpa file using CMU LMTK needed a special flag for memory calculation because of the huge number of words.

After that, generating the .lm.DMP file used in Sphinx4 was aborted with error message saying that the number of unigrams exceeded 65,535. It was impossible to generate the binary file. More attempts to generate language model using unigrams and bigrams resulted in the same problem. Unfortunately, sphinx_lm_convert tool within the CMU LMTK does not allow more than 65,535 unigrams.

Due to the limitation on the number of unigrams to 65,535, and the complexity of the work required to change the source code of the CMU LMTK and the Sphinx4 library, which would allow us to work with bigger models, we restricted ourselves to the number of unigrams ≤ 65,535.

The next implementations of enlarged language models were based on dictionaries of 30,000 and 45,000 words, respectively. As with the previous model containing 15,000 words, we tested these models without any pondering, which resulted in the same issue of incorrect recognition of words at the start of utterances. In order to solve this problem, we pondered unigrams and trigrams logarithmically as before.

After testing such models, we concluded that with linear pondering of 10^+ unigrams, 10^+ bigrams without attestation in 10^+ trigrams, and 10^+ trigrams, we can get more accurate speech recognition than with the models of logarithmically pondered unigrams and trigrams. Development of linearly pondered language models is still in progress.

The CMU LMTK limits mean that we cannot achieve 95% coverage of Croatian word usage for now, but a coverage exceeding 85% seems easily feasible. Sometime in the future it might be necessary to develop systems able to work with larger dictionaries and language models. Until then, our efforts have to be focused on improving WERs in language models within the CMU Sphinx dictionary limits.

VI. Conclusion

The results of speech recognition for our ASR system using Sphinx4 are relatively good when compared to those reported for other Croatian systems so far, especially because we have a speaker-independent system, which can easily cope with large vocabularies. Although we have already achieved good results, there is still plenty of room for improvements, from lowering WER to increasing the size of vocabulary which the system can operate with. Encouraging results reached by the rapid prototyping approach can serve as a starting point for future development of Croatian LVCSR systems able to meet public needs.

Several tests confirmed that our initial hypothesis about recording audio files in acoustically imperfect conditions, according to the original intention of making our system more robust in a real-life application, was wrong. In the future, we intend to record the training data in good acoustical conditions, i.e., without too much noise. The noise can be added manually, when needed, afterwards. By doing that, we could keep the acoustic model "clean", which would ultimately enable our system to achieve WERs much lower than presented in the paper. We believe that the handicapping of our training set for acoustic modeling with too much noise was the main reason why our LVCSR system could not achieve WERs fewer than 20% on average.

References

[1] A. W. Black et al., "TOUNGES: Rapid development of a speech-to-speech translation system," Proc. 2nd Int. Conf. on Human Language Technology Research, Morgan Kaufmann Publishers, 2002, pp. 183-186.

[2] Carnegie Mellon University (USA), Sphinx-4: A speech recognizer written entirely in the Java programming language, http://cmusphinx.sourceforge.net/sphinx4/, 15.07.2103.

[3] T. Tonžetić, Automatic speech recognition for Croatian language using Sphinx3 (in Croatian), M.S. Thesis, FER, Zagreb Univ., Croatia, 2005.

[4] S. Martinčić-Ipšić, M. Pobar, and I. Ipšić, "Croatian large vocabulary automatic speech recognition," Automatika, 2011, vol. 52, no. 2, pp. 147-157.

[5] Cambridge University (UK), HTK speech recognition toolkit, http://htk.eng.cam.ac.uk/, 16.07.2103.

[6] J. Vojtko, J. Kacur, and G. Rozinaj, "The training of Slovak speech recognition system based on Sphinx 4 for GSM networks," Proc. 49th Int. Conf. ELMAR, Zadar, Croatia, 2007, pp. 147-150.

[7] A. Janicki and D. Wawer, "Automatic speech recognition for Polish in a computer game interface," Proc. Federated Conference on Computer Science and Information Systems (FedCSIS 2011), September 2011, pp.711-716.

[8] X. Huang, A. Acero, and H.-W. Hon, "Spoken language processing," in Spoken Language Processing: A Guide to Theory, Algorithm and System Development, Prentice Hall, 2001.

[9] L. R. Rabiner and B. H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.

[10] J. P. Mallory and D. Q. Adams (eds.), Encyclopedia of Indo-European Culture, Fitzroy Dearborn Publishers, 1997.

[11] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," IEEE ASSP Magazine, January 1986, pp.4-15.

[12] M. Federico, "Bayesian estimation methods for n-gram language model adaptation," Proc. 4th Int. Conf. on Spoken Language Processing, Philadelphia, PA, USA, 1996, vol. 1, pp. 240-243.

[13] N. D. Warakagoda, A hybrid ANN-HMM ASR system with NN-based adaptive preprocessing, M.S. Thesis, Norges Tekniske Høgskole, Trondheim Univ., Norway, 1994.

[14] R. A. Cole et al. (eds.), Survey of the State of the Art in Human Language Technology, Cambridge Univ. Press, 1997.

[15] B. Dropuljić and D. Petrinović, "Development of acoustic model for Croatian language using HTK," Automatika, 2010, vol 51, no. 1, pp. 79-88.

[16] Š. Dembitz, B. Blašković, and G. Gledec, "Croatian language n-gram system," Frontiers in Artificial Intelligence and Applications, IOS Press, 2012, vol. 243, pp. 696-705.

[17] C. Manning and H. Schütze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

[18] Š. Dembitz, M. Randić, and G. Gledec, "Advantages of online spellchecking: a Croatian example," Software – Practice & Experience, 2011, vol. 41, no. 11, pp. 1203-1231.

[19] M. Radonić, Measuring word error rate in Croatian continuous speech recognition (in Croatian), B.S. Thesis, FER, Zagreb Univ., Croatia, 2013.

[20] FER, Zagreb Univ. (Croatia), HascheckVoice: The Croatian academic speech synthesizer, http://hascheck.tel.fer.hr/voice/, 16.07.2013.