Slovak Automatic Dictation System for Judicial Domain

Milan Rusko[1(✉)], Jozef Juhár[2], Marián Trnka[1], Ján Staš[2],
Sakhia Darjaa[1], Daniel Hládek[2], Róbert Sabo[1], Matúš Pleva[2],
Marián Ritomský[1], and Martin Lojka[2]

[1] Institute of Informatics, Slovak Academy of Sciences,
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
{milan.rusko,marian.trnka,utrrsach,robert.sabo,
marian.ritomsky}@savba.sk
[2] Department of Electronics and Multimedia Communications,
Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Park Komenského 13, 042 00 Košice, Slovakia
{jozef.juhar,jan.stas,daniel.hladek,matus.pleva,
martin.lojka}@tuke.sk

Abstract. This paper describes the design, development and evaluation of the Slovak dictation system for the judicial domain. The speech is recorded using a close-talk microphone and the dictation system is used for on-line or off-line automatic transcription. The system provides an automatic dictation tool in Slovak for the employees of the Ministry of Justice of the Slovak Republic and all the courts in Slovakia. The system is designed for on-line dictation and off-line transcription of legal texts recorded in acoustical conditions of typical office. Details of the technical solution are given and the evaluation of different versions of the system is presented.

Keywords: Automatic speech recognition · Slovak language · Judicial domain

1 Introduction

Dictation systems for major world languages have been available for years. The Slovak language, a Central European language spoken by a relatively small population (around 5 million), suffers from the lack of speech databases and linguistic resources, and these are the primary reasons for the absence of a dictation system until recently.

We describe the design, development and evaluation of a Slovak dictation system named APD (Automatický Prepis Diktátu – *Automatic Transcription of Dictation*) for the judicial domain.

The development of the automatic transcription systems for the judicial domain is a very challenging task from the research and development point of view. (see e.g. [1]). On the other hand there is a market demand for such technologies. Court room speech transcription is considered as one of the greatest challenges for the front-end of speech recognition and the authors with the cooperation with the Ministry of Justice of the Slovak Republic decided to divide the task into three stages.

The long-term goal is to use automatic speech processing technologies to make the judicial proceedings more effective and transparent. In order to fulfill this task, the whole legal process should be recorded and the speech utterances of all the participants should be captured and stored.

It was decided to build the dictation system for personal use in a quiet room environment in the first phase. The judges will use the system for the preparation of legal documents which are now dictated to an assistant. This stage is also intended to get the users accustomed to the new technology and gain experience with its use.

The second stage will deal with the on-line dictation by the judge and prosecutor during the trial in the acoustical conditions of the judgment hall. In this stage the focus will be given on indexing of the audio recordings using speech recognition and their storage and archiving.

The third phase should lead to a systematic and mandatory use of recording and transcribing the entire court hearings and develop a methodology and infrastructure for their use.

The paper is organized as follows: Sect. 2 introduces speech databases and annotation, Sect. 3 describes building of the APD system, Sect. 4 presents the evaluation of the dictation system, and Sect. 5 closes the paper with the discussion.

2 Speech Databases and Annotation

Several speech databases were used for acoustic models training and recognizer testing during the development of the APD dictation system. The APD database is gender-balanced and contains 250 hours of recordings (mostly read speech) from 250 speakers. It consists of two parts: APD1 and APD2.

APD1 contains 120 hours of reading transcripts of court decisions, recorded in sound studio conditions. In order to comply with current Slovak legislation on the protection of personal data, all names, addresses, and some numbers had to be changed.

APD2 contains 130 hours of read phonetically rich sentences, newspaper articles, internet texts and spelled items. This database was recorded in offices and conference rooms.

The recording of APD1 and APD2 was realized using a quiet PC with an EMU Tracker Pre USB 2.0 and E-MU 0404 USB 2.0 Audio Interface/Mobile Preamp. To obtain the signal from the different types of microphones (for future flexibility in creating acoustic models for various applications) the audio signal was simultaneously recorded to three channels with four sound tracks in total:

1. Close-talk channel using Sennheiser ME3 headset microphone with Sennheiser MZA 900 P In-Line preamplifier
2. Table microphone channel using Rode NT-3 microphone
3. Dictaphone channel that used PHILIPS Digital Voice Tracer LFH 860/880 dictaphone with its built in stereo microphone (occupying two audio tracks).

The recordings in the first two channels have 48 kHz sampling frequency and 16 bit resolution. They were later down-sampled to 16 kHz for training and testing.

The APD database was extended by the "Parliament" database which contains 96 hours of recordings realized in the main conference hall of the Slovak Parliament using conference "goose neck" microphones [2]. The sampling frequency was 48 kHz in 256 kbps CBR stereo AC-3 format. The recordings were later converted to 16 kHz mono PCM and resolution of 16 bit for training and testing. The database consists of annotated spontaneous and read utterances of 142 speakers and is not balanced for gender. In accordance with the gender structure of the Slovak Parliament it has 90 % male speakers and 10 % of female speakers.

The databases were annotated by our team of trained annotators (university students) using the Transcriber annotation tool [3] slightly adapted to meet our needs. The annotated databases underwent the second round of annotation focused on checking and fixing mistakes of the first round of annotation. The annotation files were automatically checked for typos using a dictionary-based algorithm and for correctness of event annotations from closed predefined set. The dictionary based algorithm used a dictionary with alternative pronunciations, and a high Viterbi score of forced alignment was taken to indicate files with bad annotation. Roughly 5-10 % of the files were identified as candidates for having bad annotation, and most of them really contained mistakes. The spelling of numbers was automatically checked and mistakes were corrected.

3   Building the APD System

The software solution for the APD system is designed to contain 3 layers (see Fig. 1). The CORE layer consists of the speech recognition system which communicates with the VIEW layer through the CONTROLLER layer. Such a modular design allowed us to experiment with more cores within the APD system.

Two cores were investigated. The first, in-house weighted finite state transducer (WFST) based speech recognizer, derived from the Juicer ASR decoder [4], and the second, the Julius speech recognizer [5, 6], was used mainly as our reference speech recognition engine.
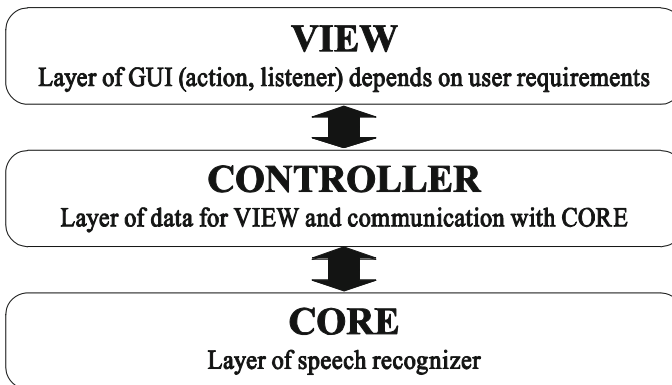

Fig. 1. The design of the APD system

3.1   Acoustic Modeling

We used a triphone mapped HMM system described in detail in [7]. Here, we briefly introduce the acoustic modeling that also has been used in the APD system.

The process of building a triphone mapped HMM system has 4 steps:

1. Training of monophones models with single Gaussian mixtures.
2. The number of mixture components in each state is incremented and the multiple GMMs are trained.
3. The output state distributions of the monophones are cloned for all triphones, triphone mapping for clustering is applied.
4. The triphone tied system is trained.

The models from Step 1 were used for the calculation of acoustic distances. The acoustic distance AD of two phonemes i and j was calculated as:

$$AD(i,j) = \sqrt{\frac{1}{V_f} \sum_{k=1}^{V_f} \frac{\left(\mu_{ik} - \mu_{jk}\right)^2}{\sigma_{ik}\sigma_{jk}}}$$

where V_f is the dimensionality of feature vector f, μ and σ are means and variances, respectively. The distance is calculated for each emitting state, resulting in i–j–1, i–j–2 and i–j–3 values for the phoneme pair (considering conventional 5 states HMM model of a phoneme, where 0th and 4th are non-emitting states).

The triphone distance TD between two triphones is defined by the following equation:

$$TD(P_a - P + P_b, P_c - P + P_d) = w_{left}AD(P_a - P_c - 3) + w_{right}AD(P_b - P_d - 1)$$

where w are context weights of the phoneme P. Then map each triphone m from the list of all possible triphones (approx. 87 k triphones), which includes unseen triphones as well, on the closest triphone $n*$ from the list of selected triphones with the same basis phoneme using metrics:

$$n^* = TriMap(m) = \arg \min_{n}(TD(m, n))$$

where $n \in \{n_1, \ldots, n_N\}$, selects N most frequent triphones in the training database (usually from 2000 to 3500).

This triphone map is then applied in Step 3 instead of conventional decision tree based state clustering, and retrained in Step 4.

Adaptation of Acoustic Models. For speaker adaptation we considered these adaptation methods and their combinations: MLLR, semi-tied covariance matrices, HLDA, and CMLLR. In our experiments we tried to choose a method that is able to improve recognition accuracy even if only a small-sized adaptation data is available. The second requirement was the simplicity of the adaptation process due to computational complexity.

For our experiments, the left-to-right three tied-state triphone HMM with 32 Gaussians was trained on recordings of parliamentary speech database and used for

adaptation [8]. A part of Euronounce bilingual database [9] containing phonetically rich sentences in the speakers' mother tongue was used for the adaptation experiments.

Adapted models were evaluated on the continuous speech with large vocabulary (>420 k words) and the achieved results are illustrated in Table 1. As we can see from the table the highest accuracy improvement with small adaptation data (50 sentences, 5 minutes of speech or less) was achieved with Maximum Likelihood Linear Regression [10] for male as well as female speakers [11].

Due to this fact we have chosen MLLR as the most suitable adaptation method for implementation. In APD LVCSR system the supervised MLLR adaptation was implemented using the predetermined regression classes.

Table 1. Evaluation of acoustic models with 4, 8, 16, and 32 Gaussians per state adapted to a male speaker

Man speaker	Basic AM WER [%]	Semi-tied WER [%]	Semi-tied + HLDA WER [%]	MLLR WER [%]	CMLLR + MLLR WER [%]
4 mix	31.06	21.12	14.31	19.08	21.39
8 mix	25.96	24.39	13.24	14.57	16.38
16 mix	21.36	34.69	14.87	10.96	12.18
32 mix	17.89	46.53	16.41	11.12	11.50

Figure 2 illustrates how MLLR was implemented. As can be seen, we focused on mean vectors adaptation while variances of mixtures stayed the same as in the original un-adapted acoustic model. The algorithm iterated through every class with assigned adaptation vectors of every adaptation sentence.

Firstly, we have calculated the score of every mixture for the aligned state. Then we were able to calculate γ as a normalized weight for every mixture of state and matrix X as the sum of multiplications of γ, feature vector O and extended mean vector Eu for all previous mixtures. The matrix Y is calculated as a sum of multiplication of the extended mean vectors Eu_{max} of mixture with highest score b_{max} within state of HMM. In this case the γ was equal to one.

After the matrices X and Y were calculated for all classes, it was possible to calculate transformation matrix W as a multiplication of matrix X and inverted matrix Y for that particular class.

When transformation matrices are known for every class it is possible to adapt all mean vectors of the acoustic model using an appropriate transformation matrix for a given mixture.

3.2   Language Modeling

Text Corpora. The main assumption in the process of creating an effective language model (LM) for any inflective language is to collect and consistently process a large amount of text data that enter into the process of training LM.

```
for all classes(i)
    for all adaptfiles
        for all vectors
           total score=calculate score for all mix
           for all mixtures of assigned state
               γ=score[mix]/total score
               Compute X_i+=γ.O.Eu
           end
           find mixture with highest score (b_max)
               Compute Y_i+=γ.Eu_max.Eu_max
        end
    end
end
for all classes(i)
    inverse Y_i
    Compute W_i=X_i.Y_i^{-1}
end
for all classes(i)
    for all mixtures
        adaptMean=W_i.Eu
    end
end
```

Fig. 2. Pseudo code of implemented MLLR

Text corpora were created using a system that retrieves text data from various Internet pages and electronic documents that are written in Slovak [12]. Text data were then normalized by additional processing such as word tokenization, sentence segmentation, numerals transcription, abbreviations expanding, and others.

The system for text gathering also includes application of constraints such as filtering of grammatically incorrect words by spelling check, duplicity verification of text documents, and other constraints.

The text corpus containing more than 2 billion of Slovak words in 120 million sentences was split into several different domains, as we can see in Table 2 [13].

Table 2. Statistics of text corpora

Text corpus	# sentences	# tokens
Web corpus	50 694 708	748 854 697
Broadcast news	36 326 920	554 593 113
Judicial corpus	18 524 094	565 140 401
Corpus of fiction	8 039 739	101 234 475
Unspecified text	4 071 165	55 711 674
Annotations	485 800	4 434 217
Development set	1 782 333	55 163 941
Together	**119 924 759**	**2 085 132 518**

Vocabulary. The vocabulary used in modeling of the Slovak language was subsequently selected using standard methods based on the most frequent words in the text corpora mentioned above and maximum likelihood approach for the selection of specific words from the judicial domain [13]. The vocabulary was extended with specific proper nouns and geographic entities in the Slovak Republic.

We have also proposed an automatic tool for generating inflective word forms for names and surnames which are used in language modeling using word classes based on their grammatical category [14]. The final vocabulary contains 325 555 unique word forms, 22 grammatically dependent classes with 97 678 proper nouns and geographic entities and 5 tags for noise events (474 450 pronunciation variants).

Language Model. The process of building a language model for the Slovak language consists of the following steps.

First, the statistics of trigram counts from each of the domain-specific corpora are extracted. From the extracted n-gram counts the statistics of counts-of-counts are computed for estimating the set of Good-Turing discounts needed in the process of smoothing LMs. From the obtained discounts the discounting constants used in smoothing LMs by the modified Kneser-Ney algorithm are calculated.

Particular trigram domain-specific LMs are created using SRILM Toolkit [15] with the vocabulary and smoothed by the modified Kneser-Ney or Witten-Bell algorithms. In the next step, the perplexity of each domain-specific LM for each sentence in the development data set from the field of judiciary is computed. Perplexity is a standard measure of quality of LM which is defined as the reciprocal value of the (geometric) average probability assigned by the LM to each word in the evaluated (development) data set.

From the obtained collection of files, the parameters (interpolation weights) for individual LMs are computed by the minimization of perplexity using an EM algorithm. The final LM adapted to the judicial domain is created as a weighted combination of individual domain-specific trigram LMs combined with linear interpolation.

Finally, the resulting model is pruned using relative entropy-based pruning in order to use it in the real-time application in domain-specific task of Slovak LVCSR [13].

3.3   User Interface

The text output of the recognizer is sent to a text post-processing module. Some of the post-processing functions are configurable though the user interface. After launching the program, the models are loaded and the text editor (Microsoft Word) is opened. The dictation program is represented by a tiny window with **start/stop, launch Word, microphone sensibility,** and **main menu** buttons (see Fig. 3 from APD 0.9 version of the system). There is also a sound level meter placed in line with the buttons.

The main menu contains **user profiles**, audio setup, program settings, off-line transcription, user dictionary, speed setup, help, and "about the program" submenus. The user profiles submenu gives an opportunity to create a new user profile (in addition to the general "**male**" and "**female**" profiles that are available with the installation). The procedure of creating a new profile consists of an automatic microphone sensitivity setup procedure and reading of 60 sentences for the acoustic
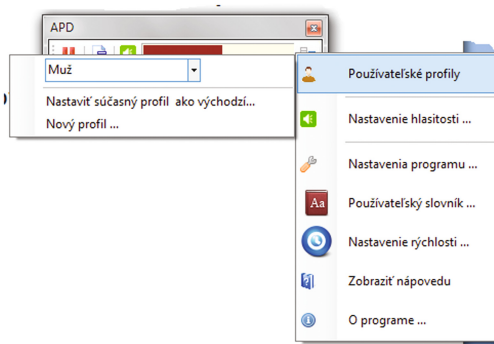
Fig. 3. APD interface screenshot

model adaptation. The new profile keeps information about the new user program and audio settings, user dictionary etc. The **audio setup** submenu opens the audio devices settings from the control panel. The **program settings** submenu makes it possible to:

- create user-defined corrections/substitutions (abbreviations, symbols, foreign words, named entities with capital letters, etc.)
- define voice commands
- insert user defined MS Word files using voice commands.

The **off-line transcription** mode makes it possible to transcribe recorded speech files into text. The **user dictionary** submenu allows the user to add and remove items in the dictionary. The words are inserted in the orthographic form and the pronunciation is generated automatically. The **speed setup** submenu allows the user to choose the best compromise between the speed of recognition and its accuracy. The **help** submenu shows the list of basic commands and some miscellaneous functions of the program.

The dictation program has a **special mode for spelling**. The characters are entered using the Slovak spelling alphabet (a set of words which are used to stand for the letters of an alphabet). The numerals, roman numerals and punctuation symbols can be inserted in this mode as well. This mode uses its own acoustic and language models. Large amount (approx. 60 k words) of Slovak proper nouns is covered by the general language model, but the user can also switch to a special mode for proper names, that have a special, enriched "name vocabulary" (approx. 140 k words) and a special language model.

4   Evaluation

We evaluated core speech recognizers of the APD systems, and the overall performance of dictation in the judicial domain.

4.1 Comparison of Core ASR Systems

The design of the system allowed us to use more speech recognition engines. The first one, the in-house speech recognition system, has been derived from the Juicer ASR system. All HTK dependencies were removed and considerable speed enhancing improvements were implemented, such as fast loading of acoustic models, and fast model (HMM state) likelihood calculation.

We implemented the highly optimized WFST library for transducer composition, which allowed us to build huge transducers considerably faster (WFST composition was 4 times faster) than with the standard AT&T FSM library. See [16, 17] for more details.

The second one, the Julius speech recognition system was used as the reference ASR system.

The comparison was performed on Parliament task (using the training set of utterances from Parliament database, and the 3 hours testing set not included in the training set.).

We can also view this comparison in light of the comparison between a conventional ASR system (Julius) and a WFST based ASR system. The dictionary used in the experiment contained about 100 k words. Table 3 shows the results.

Table 3. ASR core systems comparison

Decoder	Task	n-gram	WER [%]
Julius	Parliament 100 k	3	17.5
In-house	Parliament 100 k	3	17.2
In-house	Parliament 100 k alt. silences (*sp*)	3	16.4
In-house	Parliament 100 k	4	17.2

We see from the comparison that the performance of WFST-based (in-house) decoder is very similar to the conventional sophisticated decoder (Julius, forward search with bigram LM and backward speech with reversed trigram LM). We were able to achieve a performance gain with easy manipulation of the search network – WFST. Adding alternatives to every silence in the graph, null transition and *sil* transition to every *sp* transition, we observed significant improvement.

However, this improvement was accompanied by bigger transducers. Building such a transducer was a tractable problem for this smaller vocabulary (100 k words), however for real tasks with 433 k words we had in the APD system, the building and the use of such huge transducers was not possible.

4.2 Performance of the APD System

The testing set of 3.5 hours consists of recordings taken from APD2 database. It was not included in the training set. Table 4 shows the results.

Two gender profiles are available within the APD system. Each profile used gender dependent acoustic models, and Table 4 presents the results of male models.

Table 4. Evaluation of the APD dictation system

Acoustic model	WER [%]
APD1 + APD2	7.72
APD2 + Parliament	5.83
APD1 + APD2 + Parliament	5.26
Male APD1 + APD2 + Parliament	5.38
Male MPE APD1 + APD2 + Parliament	5.35

The testing set contained around 2 hours of male recordings, the subset of full testing set.

Discriminative modeling was also used. Both MMI and MPE models were trained. While on the Parliament task more than 10 % relative improvement was observed, on the APD task we got only slight improvement.

5   Conclusions

We presented the design and development of the first Slovak dictation system for the judicial domain. Using in-domain speech data (the APD speech database) and text data, good performance has been achieved.

Last year the APD system has been installed and used by 1200 persons (judges, court clerks, assistants and technicians) at different institutions belonging to the Ministry of Justice. During this period the activation module of the system registered 700 automatic preinstalled version (using special image on new computers for this purpose), 410 email activations using court IT specialist installation and 94 telephone activations by judges or other court end-users. There was a telephone and email help desk provided by the development team and a web questionnaire for end-users realized inspired by [18]. The results of the field tests are analyzed and the results will be published soon.

We can conclude that the first version of APD is already used at the organizations belonging to the Ministry of Justice of the Slovak Republic. For dissemination of the APD system a product leaflet was distributed to the courts, the end-user and IT specialist training was prepared and finally an internal circular letter from the Ministry of Justice to the courts was sent to inform the end-users about the possibility to speed up the court proceedings.

Further we plan to extend the APD system with automatic speech transcription used directly in the court rooms, and later with speech transcription of whole actions in court, similarly as described in Polish and Italian speech transcription systems for judicial domain [1]. The system will be designed to cope with factors such as distant talk microphones, cross channel effects and overlapped speech.

References

1. Lööf, J., Falavigna, D., Schlüter, R., Giuliani, D., Gretter, R., Ney, H.: Evaluation of automatic transcription systems for the judicial domain. In: Proceedings of the IEEE Workshop on Spoken Language Technology, San Francisco, CA, USA, pp. 194–199 (2010)
2. Darjaa, S., Cerňak, M., Beňuš, Š., Rusko, M., Sabo, R., Trnka, M.: Rule-based triphone mapping for acoustic modeling in automatic speech recognition. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS (LNAI), vol. 6836, pp. 268–275. Springer, Heidelberg (2011)
3. Barras, C., Geoffrois, E., Wu, Z., Liberman, M.: Transcriber: development and use of a tool for assisting speech corpora production. Speech Commun. Speech Annot. Corpus Tools **33**(1–2), 5–22 (2001)
4. Moore, D., Dines, J., Doss, M.M., Vepa, J., Cheng, O., Hain, T.: Juicer: a weighted finite-state transducer speech decoder. In: Renals, S., Bengio, S., Fiscus, J.G. (eds.) MLMI 2006. LNCS, vol. 4299, pp. 285–296. Springer, Heidelberg (2006)
5. Lee, A., Kawahara, T., Shikano, K.: Julius – an open source real-time large vocabulary recognition engine. In: Proceedings of EUROSPEECH 2001, Aalborg, Denmark, pp. 1691–1694 (2001)
6. Lee, A., Kawahara, T.: Recent development of open-source speech recognition engine julius. In: Proceedings of the 2009 Asia-Pacific Signal and Information Processing Association, Annual Summit and Conference, APSIPA ASC 2009, Sapporo, Japan, pp. 131–137 (2009)
7. Darjaa, S., Cerňak, M., Trnka, M., Rusko, M., Sabo, R.: Effective triphone mapping for acoustic modeling in speech recognition. In: Proceedings of INTERSPEECH 2011, Florence, Italy, pp. 1717–1720 (2011)
8. Papco, M., Juhár, J.: Comparison of acoustic model adaptation methods and adaptation database selection approaches. J. Electr. Electron. Eng. **3**(1), 147–150 (2010). ISSN: 1844–6035
9. Jokisch, O., Wagner, A., Sabo, R., Jackel, R., Cylwik, N., Rusko, M., Ronzhin, A., Hoffman, R.: Multilingual speech data collection for the assessment of pronunciation and prosody in a language learning system. In: Proceedings of the 13th International Conference Speech and Computer, SPECOM 2009, St. Petersburg, Russia, pp. 515–520 (2009)
10. Leggetter, C.J., Woodland, P.C.: Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Comput. Speech Lang. **9**(2), 171–186 (1995)
11. Papco, M., Juhár, J.: MLLR adaptation of speaker independent acoustic model to a new speaker in LVCSR. In: Proceedings of the 7th International Workshop on Digital Technologies, DT 2010, Žilina, Slovakia, pp. 1–4 (2010)
12. Hládek, D., Staš, J.: Text Mining and Processing for Corpora Creation in Slovak Language. J. Comput. Sci. Control Syst. **3**(1), 65–68 (2010)

13. Juhár, J., Staš, J., Hládek, D.: Recent progress in development of language model for Slovak large vocabulary continuous speech recognition. In: New Technologies: Trends, Innovations and Research, InTech, Rijeka, Croatia, pp. 261–276 (2012)
14. Staš, J., Hládek, D., Juhár, J.: Morphologically motivated language modeling for Slovak continuous speech recognition. J. Electr. Electron. Eng. **5**(1), 233–236 (2012)
15. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: Proceedings of ICSLP 2002, Denver, Colorado, USA, pp. 901–904 (2002)
16. Lojka, M., Juhár, J.: Fast construction of speech recognition network for Slovak language. J. Electr. Electron. Eng. **3**(1), 111–114 (2010)
17. Juhár, J., Lojka, M.: Fast construction of static speech recognition network with low memory requirement. In: Proceedings of the International Conference on Applied Electrical Engineering and Informatics, AEI 2011, Venice, Italy, pp. 163–166 (2011)
18. Alapetite, A., Boje Andersen, H., Hertzum, M.: Acceptance of speech recognition by physicians: a survey of expectations, experiences, and social influence. Int. J. Hum Comput Stud. **67**(1), 36–49 (2009)