Fast Keyword Spotting in Telephone Speech

Jan NOUZA, Jan SILOVSKY

SpeechLab, Faculty of Mechatronics, Technical University of Liberec, Studentska 2, 46117 Liberec, Czech Republic

jan.nouza@tul.cz, jan.silovsky@tul.cz

Abstract. *In the paper, we present a system designed for detecting keywords in telephone speech. We focus not only on achieving high accuracy but also on very short processing time. The keyword spotting system can run in three modes: a) an off-line mode requiring less than 0.1xRT, b) an on-line mode with minimum (2 s) latency, and c) a repeated spotting mode, in which pre-computed values allow for additional acceleration. Its performance is evaluated on recordings of Czech spontaneous telephone speech using rather large and complex keyword lists.*

Keywords

Speech processing, keyword spotting, speech decoder, HLDA transformation.

1. Introduction

Keyword spotting (KWS) has become an important branch of speech technology. It is applied mainly in situations where a large amount of spoken documents must be searched to learn whether they contain some specific words. The fast detection of these words (and information about their exact location) eliminates a lot of human work in such tasks like audio data mining, named entity search, and, in particular, in the state security domain.

In general, there are two main approaches used for keyword spotting [1], [2]. The most natural one consists in performing complete transcription of the documents (using the best available large-vocabulary speech recognition system) first and then detecting the words of interest in the text version of the documents. Obviously, this approach works well in situations where a) speech quality and speaking style allow the recognizer to produce text with minor errors only, b) the searched words are in the recognizer's vocabulary, c) a longer processing time does not matter. (A good example is, e.g. data mining in broadcast news [3]).

In typical security tasks, however, these assumptions often do not apply. Here, one of the major types of analyzed documents is a telephone call. It is a narrow-band, low-quality audio signal with speech that is usually informal (with respect to lexicon, grammar and pronunciation) and highly spontaneous with frequent artifacts like hesitations, repeated words or interruptions. For this type of spoken data, an approach that uses smaller vocabularies (usually made of the searched words only) and so called fillers (that capture and cover the rest of speech) is more suitable [4].

In this paper, we describe the system we developed for detecting keywords in telephone conversation. The main requirements were as follows:

- Primary language of the calls is Czech, though foreign words (especially names) can occur and can be searched.
- The lists of searched words may include hundreds of words and since Czech is an inflected language, the actual list size can grow up to thousands of items.
- The performance should be as high as possible, allowing individual setting to prefer either higher detection rate or lower false alarm rate.
- The processing time should be as short as possible (a fraction of real time (RT), possibly < 0.1 RT).

In the design, we applied the approach based on the word and filler model, which is the only one that can fulfill the last mentioned requirement. Moreover, we focused on proposing such a solution than can run not only in the off-line mode, but also in an on-line mode (e.g. for direct monitoring of a telephone line with an immediate alarm triggered by one of the list words). In the design, we have included also an option that makes the repeated search in the same audio data faster. This is possible by pre-computing and storing some of the values used by the speech decoder.

2. KWS System and Its Decoder

The KWS system consists of several basic modules. The audio input module performs initial preprocessing of speech signal that can be stored (or provided from a line) in different formats. The output from this module is a classic 8 kHz 16-bit PCM-coded signal. In case of a stereo-recorded call, two separated signals are created. The next module makes signal parameterization, computes feature vectors, normalizes them inside a sliding window, and also uses them to decide the gender of the speaker. The feature vectors and the information about the gender are passed to

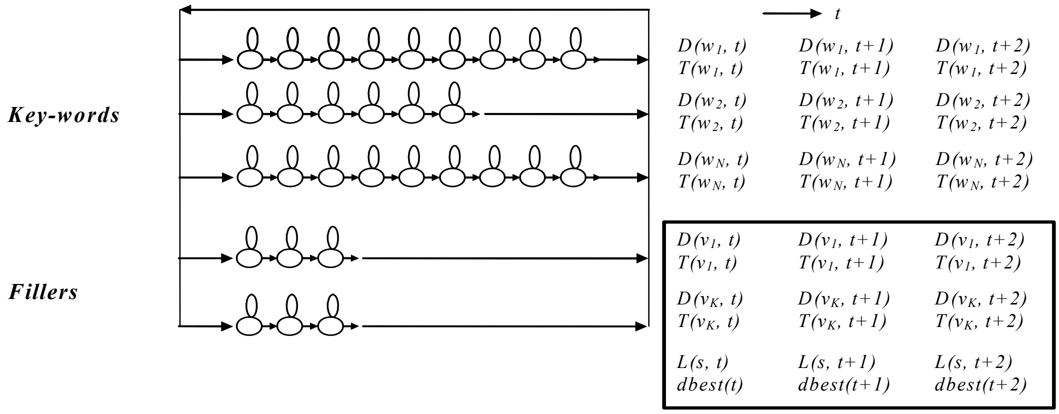
Key-words
Fillers
t
D(w1, t) D(w1, t+1) D(w1, t+2)
T(w1, t) T(w1, t+1) T(w1, t+2)
D(w2, t) D(w2, t+1) D(w2, t+2)
T(w2, t) T(w2, t+1) T(w2, t+2)
D(wN, t) D(wN, t+1) D(wN, t+2)
T(wN, t) T(wN, t+1) T(wN, t+2)
D(v1, t) D(v1, t+1) D(v1, t+2)
T(v1, t) T(v1, t+1) T(v1, t+2)
D(vK, t) D(vK, t+1) D(vK, t+2)
T(vK, t) T(vK, t+1) T(vK, t+2)
L(s, t) L(s, t+1) L(s, t+2)
dbest(t) dbest(t+1) dbest(t+2)

Fig. 1. Network of key-words (w) and fillers (v). Denoted are word-end accumulated scores D, starting times T, likelihoods L, and values *dbest*. The values in the rectangle are word independent and can be used in repeated runs with different words lists.

the decoder. It selects the appropriate acoustic model, performs speech decoding, provides hypotheses about the presence of keywords in the signal, and quantifies their scores. The last module takes these hypotheses, compares them to pre-set thresholds and produces an output list with detected words, their time markers and confidence values.

In the next text, we describe the decoder, which is the core component of the system, in more details. We focus mainly on those parts that have been optimized for speed.

2.1 KWS Decoder

The decoding is based on the well-known Viterbi algorithm. We have utilized its fast implementation created for the LVCSR system [5]. Hence, the KWS system can be used even for list with thousands of keywords.

The decoder operates with a looped network of units u that are either keywords w or fillers v. Both are handled in the same way. The fillers are represented by models of all 41 Czech phonemes and 7 non-speech events (silence and various noises). The words use the same phoneme models. These are 3-state context-independent HMMs with a large number of Gaussians per state. In our implementation, we omit transition probabilities, which makes computation faster without any noticeable impact on the accuracy.

The elementary operation in the Viterbi decoder is the propagation of the accumulated scores to adjacent states. At each time (frame) t, new accumulated score d is computed for each state s of unit u by adding log likelihood L of feature vector $\mathbf{x}(t)$ to the higher of the scores in the predecessor states:

$$d(u,s,t) = L(s,\mathbf{x}(t)) + \underset{i=0,1}{Max}[d(u,s-i,t-1)]\,. \tag{1}$$

To decode the sequence of units, we are interested mainly in scores D achieved at time t in last states s_e of the units

$$D(u,t) = d(u,s_e,t)\,. \tag{2}$$

Furthermore, we need to register time $T(u,\ t)$ when the given instance of unit u started.

To close the loop, at each time t we compute value

$$D_{best}(t) = \underset{u}{Max}[D(u,t)] \tag{3}$$

and propagate it to initial states s_b of all units:

$$\begin{aligned} d(u,s_b,t) &= L(s_b,\mathbf{x}(t)) \\ &\quad + Max[D_{best}(t-1), d(u,s_b,t-1)] \end{aligned} \tag{4}$$

To get acoustic score S of unit u we have to subtract the two accumulated scores:

$$S(u,t) = D(u,t) - D_{best}(T(u,t)-1) \tag{5}$$

For each word w, we have to compare its score $S(w,t)$ with score $S_f(v_{conc},t)$ that would be achieved by the best concatenation of fillers starting in time $T(w,\ t)$ and ending in time t. Basically, this score can be computed by applying the Viterbi algorithm to the given time span and to the filler models only. (In practice, it can be approximated by applying (5) to the best filler model ending in time t.) Then, we define normalized acoustic score S_N as:

$$S_N(w,t) = S(w,t) / S_f(v_{conc},t)\,. \tag{6}$$

This normalized score will reach its maximum value 1 only if keyword w gets the same acoustic score as the concatenation of the fillers made of the word's phonemes. In this case, we can be sure that the keyword was detected correctly. In other cases, $S_N < 1$ and the probability of the correct detection decreases. The proper threshold for rejecting/accepting a keyword must be set experimentally on development data.

2.2 Speed Optimization of Decoder

It is known that the major bottleneck in the decoding procedure is the computation of likelihoods L occurring in (1). In a typical KWS system, this may take up to 90 % of the total processing time. In our system, we use the fast implementation whose basic ideas are described in [6]. Instead of summing contributions of all the Gaussians in the state, we take the likelihood of the best one, and instead of summing over all the features in the innermost loop, we apply an early break whenever it is possible. This scheme reduces the likelihood computation to almost one half.

In [6] we also describe our implementation of the efficient beam search, whose thresholds for each frame *t* are derived from values d_{best}:

$$d_{best}(t) = \underset{u,s}{Max}\, d(u,s,t) \tag{7}$$

As we show in Section 4, the off-line version of the KWS system that utilizes the above optimizations can run faster than 0.1 RT. Though, in a special case, the execution time can be reduced even further. It is the case when the same audio data is searched repeatedly. In the security domain, it happens quite often that archived records are analyzed not only once but several times, and usually with different keyword lists.

In this special case, we can save a large portion of repeated computation if we store the values that are keyword independent and – at the same time - critical for the decoder's performance. The list of these values is highlighted in Fig.1. It consists of values *D* and *T* (for fillers), likelihoods *L* and value d_{best} – for each frame of speech. Usually, due to pruning, not all of them are actually computed and thus not all of them need to be stored. Even if we store all, the maximum required space would not be large: 48 + 48 + 48 x 3 + 1 = 241 numbers (964 bytes) per frame. Compared to the classic PCM coding (160 bytes per 10-ms-long frame), this is only 6 times more data.

If we store these pre-computed values in special files and utilize them in repeated spotting sessions, we completely eliminate computation of a) signal processing, b) likelihoods, c) fillers, and d) beam search parameters. The repeated search thus consists only in a simple Viterbi recombination and summation of existing values, and in score normalization. Our experiments showed that in this case, the KWS system performance could be 2 – 4 times faster than the standard approach. (The actual acceleration factor depends on the keyword list size.)

3. Signal Processing and Acoustic Model

In this section, we briefly describe the acoustic part of the KWS system.

3.1 Signal Processing

The features used in the system are Mel-frequency cepstral coefficients. The set of 13 MFCCs (including c0) is extracted from the signal using 25 ms window and 10 ms shift. To compensate for possible channel and speaker change effects, we employ the CMS (cepstral mean subtraction) technique. It is applied locally within a 400 frame sliding window and only the central frame is adapted. The feature vector is further augmented by the 1st and 2nd derivatives (Δ+ΔΔ). Finally, the HLDA transformation [7], [8] is applied to reduce the original 39-feature vector to a 26-feature one. This makes the decoding faster and also yields slightly higher accuracy. More details about the feature selection and comparison can be found in Section 4.

3.2 Acoustic Models

A speaker-independent (SI) and two gender-dependent (GD) acoustic models were trained on the available corpus of Czech telephone speech. This (rather heterogeneous) database contains 37.5 hours of read speech, 25.3 hours of conversational speech of radio broadcast callers and 43.8 hours of spontaneous speech. The database is well balanced with respect to the gender of speakers (52 % male, 48 % female speech). This reasonably large amount of data (more than 50 hours for each gender) allowed us to train gender-dependent acoustic models. The HLDA transformation was estimated for each model while sharing the same training data. All the 3 types of the acoustic model consist of 48 3-state HMMs with 96 Gaussian components per state.

3.3 Gender Identification

The GD models are preferred because they contribute to slightly higher recognition accuracy. Obviously, their usage requires that a proper gender identification module is included. Ours is based on Gaussian Mixture Models (GMMs) operating with the same MFCC features used for speech recognition. The system design reflects the needs to process rather long audio streams (up to several hours), in which speakers can change frequently. Hence, the gender identification is performed locally, within a 400-frame-long sliding window (4 seconds). The implementation of the system allows for switching between the 2 acoustic models for every frame without any delay. However, if the models have tendency to switch frequently (in segments shorter than 1 s), it means that the gender is not identified reliably and then the SI model is employed instead.

3.4 Enhancing the Robustness

Continuous audio streams recorded via a telephone line monitoring system contain a lot of various non-speech events, e.g. DTMF sounds, line busy tones, music in background, etc. As the KWS system tends to generate a higher number of false alarms in non-speech regions, a speech activity detector must be included. In our system, we do it by extending the gender identification module by adding a third GMM tailored to the non-speech events.

Another source of performance degradation is the over-excitation of the signal. A set of heuristic rules based on the energy of the signal in the time span occupied by the detected keyword is therefore used in the decision making strategy. This also eliminates the false alarm detections in the silence regions.

4. Experiments

4.1 Evaluation Metrics

The performance of the KWS system is evaluated by two widely used metrics – Figure of Merit (FOM) and Equal Error Rate (EER). We also use a Receiver Operating Characteristic (ROC) curve in some experiments. The ROC curve shows the trade-off between the detection rate (DR) and false alarm (FA) rate depending on the value of the decision threshold. Values DR and FA are given as

$$DR[\%] = N_{correct} / N_{kw\_occur} \times 100\,, \tag{8}$$

$$FA[1/kw/h] = N_{FA} / (Dur \times N_{kw}) \tag{9}$$

where $N_{correct}$ represents correct detections, N_{kw_occur} is the number of all occurrences of the keywords in reference transcriptions, N_{FA} is the number of false alarm detections, N_{kw} is the number of keywords, and finally Dur is the overall duration (in hours) of all test recordings. The FOM value is defined as the average value of detection rates corresponding to FA values in the range from 0 to 10. The EER value reflects the situation when the number of missed (not detected) keywords is equal to the number of incorrect detections.

4.2 Evaluation Data

A series of experiments was performed on a portion of about 2 hours of data drawn from the spontaneous speech part of the aforementioned database. These data were excluded from the training process. The test recordings were excerpts from spontaneous conversations, and each contained one utterance spoken by a single speaker. A precise, human-made and time aligned transcription was provided for each of the test recording.

Two distinct keyword sets were prepared for the evaluation. The first set (KWSET1) was used primarily for system development purposes and it was used in all the reported experiments if not stated otherwise. The set contained 570 words. These words were chosen to be rather long (6 to 15 phonemes) and mutually dissimilar (differing in at least 3 phonemes), in order to eliminate wrong evaluation caused by possible mistakes in reference transcriptions. The second keyword set (KWSET2) represents a more challenging task. Its list contains 508 shorter words (4 to 12 phonemes), some being acoustically very similar each other (e.g. “jedna”, “jedno”, “jednu”).

4.3 Tests with Different Acoustic Features

Three types of acoustic features were examined in the initial experiments – MFCC, MFCC with HLDA transformation and Perceptual Linear Predictive (PLP) [9] coefficients. In Tab. 1 we summarize the achieved results in terms of FOM and EER values and processing time. The latter is stated as a real-time factor measured on modern PC processor Intel Core2Duo E6750 (single core in use).

	FOM [%]	EER [%]	Time ×RT
39 MFCC (Δ, ΔΔ)	80.7	39.6	0.18
39 PLP (Δ, ΔΔ)	80.5	43.2	0.18
39 MFCC (Δ, ΔΔ) + HLDA	81.5	38.7	0.13

Tab. 1. Comparison of results achieved for various acoustic parameter types.

When comparing the MFCC and PLP features, we can notice a slightly better accuracy provided by the former ones. The use of the HLDA transformation yielded another small improvement in performance and also a significant reduction of processing time - about 25 % due to the lower feature vector dimension.

4.4 Processing of Long Audio Streams

In this section, we want to highlight the effect of the local application of both the CMS and the GD acoustic models. The short (sentence-long) segments used in the previous experiments do not reflect the situation when a long continuous audio stream from a telephone line is to be monitored. In practice, this type of usage is quite frequent and it is more challenging because the assumption about the same channel characteristics and a single speaker often does not apply.

Hence, to test the robustness of our system in these conditions, we created an artificial 2-hour-long stream by concatenating all the recordings used in the previous experiment. The comparison of the results presented in the first line of Tab. 2 to those in Tab. 1 clearly demonstrate that a severe degradation of the performance occurs when the CMS technique and GD acoustic model is applied globally.

In order to cope with the varying acoustic conditions in the audio stream we introduced a “floating CMS” scheme. It consists in the local application of the CMS with the cepstral mean computed within a sliding window of a fixed length. (The choice of 400 frames was found as optimal in preliminary experiments.) The application of this locally estimated CMS and the usage of the SI acoustic model yielded a reasonable performance gain, as it can be observed from the second line in Tab. 2. Though, these results were still significantly worse compared to those reported for segmented recordings. A slight improvement was achieved by the utilization of the acoustic model formed by merging the male and female model into a super-model with the double number of Gaussians. However, the best results were achieved when the same sliding window was used both for the CMS as well as for the gender identification and the proper gender model selection. In this case, the results (presented in the fourth line of Tab. 2) are almost comparable with those in Tab. 1. In Fig. 2 we also show the ROC curves for all the experiments.

	FOM [%]	EER [%]
global CMS, global GD models	62.8	57.0
local CMS, SI model	72.4	50.1
local CMS, merged GD models	72.4	48.6
local CMS, local GD models	78.3	41.9

Tab. 2. Impact of local application of CMS and local selection of GD acoustic models in processing of long streams.

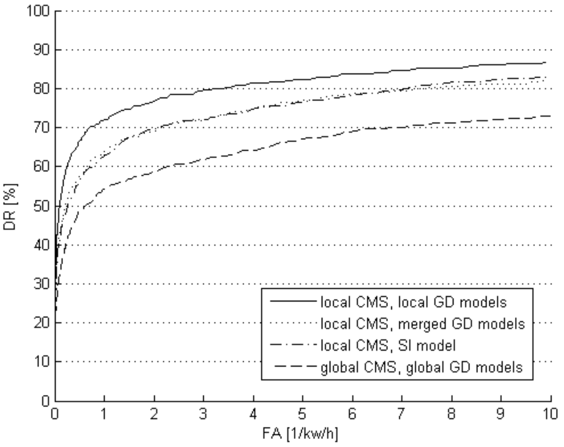

Fig. 2. Impact of local application of CMS and local selection of GD acoustic models in processing of long streams.

4.5 Speed Tuning

Tab. 3 shows the results achieved by applying the speed optimization techniques described in Sections 2.1 and 2.2. In the baseline system, we used the decoder that had been previously optimized for LVCSR tasks and which is capable of real-time operation with 300K vocabularies. By optimizing the decoder for the KWS task and by including the fast likelihood routine we were able to save more than 50 % of computation demands. Recently, with keyword lists that have size of several hundreds of words (like the sets KWSET1 and KWSET2) the complete processing time is about 0.06 RT. In the last line of Tab. 3 we also present the time needed for the repeated run of the keyword spotter in case when the selected values are pre-computed and stored as it is explained in Section 2.2.

	FOM [%]	EER [%]	Time ×RT
Baseline implementation	81.5	38.7	0.13
+ fast likelihood computation	81.1	38.5	0.06
Repeated run with pre-computed data	81.1	38.5	0.02

Tab. 3. Impact of proposed speech optimization techniques.

4.6 More Challenging Keyword List

All the previous experiments were performed using the keyword set KWSET1. Fig. 3 provides a graphical comparison of these results with those achieved for keyword set KWSET2. Here, we can observe the strong impact of the type of the searched keywords on the system performance. The KWS strategy that is based on acoustic information only can hardly distinguish between words that are phonetically very similar (or may be even homophones). These errors could be eliminated only by taking the sentence context into account in the same way as it is done in large vocabulary continuous recognition. The LVCSR approach, however, is much slower and in fact its performance is also significantly degraded in situations where spontaneous speech is transmitted by low-quality telephone line.

When we analyzed the results from the experiments with the KWSET2, we found out that the main source of errors was significantly high percentage of false alarm detections for short words (3 to 5 phonemes, many of them differing only in a single phoneme). This is because the scores for short words, computed over a short time span, are very similar, and it is not easy to set up a fixed or flexible threshold for their acceptance or rejection. So, the crucial problem of the very short words is not to detect them but to reduce the occurrence of false alarms at the same time.

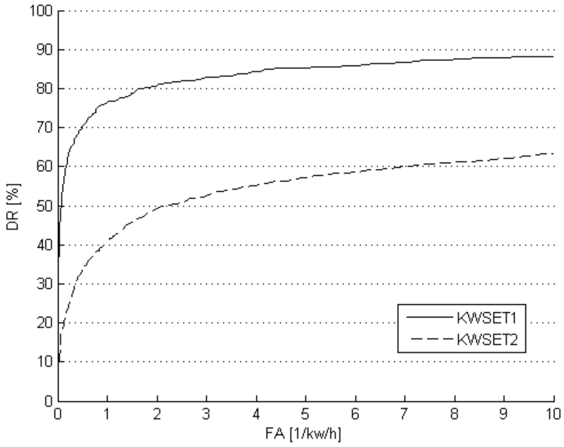

Fig. 3. Comparison of ROC curves for two sets with different types of searched words.

5. Conclusions

In this paper we present the methods used for the development of a practical keyword spotting system. The system was designed for Czech language but all its modules, except of the acoustic model trained on Czech phonemes, are language independent. We focused mainly on the optimization of speed of the system because in applications, like telephone call monitoring for state security services, short processing time is one of the main requirements.

Our system proved its capability to operate faster than 0.1 RT with a vocabulary containing about 600 keywords. We showed that in the off-line mode, its response can be further increased in situations when recordings are searched repeatedly with different keywords or different setting (e.g. with a larger or smaller beam width). In this case the system utilizes auxiliary files with pre-computed values of likelihoods, scores and time markers. Moreover, the system can be used also in an on-line mode. The signal preprocessor and the decoder are designed in the way that the detected keyword candidates can be output with a short delay after they occur. In the current implementation, this latency is 2 seconds and it is determined mainly by the size of the sliding window used for the cepstral mean normalization and gender identification.

We also demonstrate how the local application of the CMS and the local choice of the proper GD/SI model enhance the robustness of the system against varying acoustic conditions and speaker changes in continuous recordings from a telephone line.

Acknowledgements

The research described in this paper was supported by project of the Czech Ministry of Interior (project no. VD20072010B160) and by the Czech Grant Agency project no. 102/08/0707.

References

[1] ALON, G. Key-word spotting – The base technology for speech analytics. *Natural Speech Communications*, July 2005.

[2] SZOKE, I., SCHWARZ, P., MATEJKA, P., BURGET, L., FAPSO, M., KARAFIAT, M., CERNOCKY, J. Comparison of keyword spotting approaches for informal continuous speech. In *Proc. of Interspeech 2005*. Lisbon (Portugal), Sept. 2005, p. 633–636.

[3] NOUZA, J., ZDANSKY, J., CERVA, P., KOLORENC, J. A System for information retrieval from large records of Czech spoken data. *Lecture Notes in Computer Science. LNAI 4188*. Berlin, Heidelberg : Springer-Verlag, 2006, pp. 485-492.

[4] KNILL, K. M., YOUNG, S. J. Fast implementation methods for Viterbi-based word-spotting. In *Proc. of ICASSP 1996.* Atlanta (USA), 1996, p. 522-525.

[5] NOUZA, J., ZDANSKY, J., CERVA, P., KOLORENC, J. Continual on-line monitoring of Czech spoken broadcast programs. In *Proc. of Interspeech 2006*. Pittsburgh (USA), 2006, p. 1650-1653.

[6] NOUZA, J., CERVA, P., ZDANSKY, J. Very large vocabulary voice dictation for mobile. In *Proc. of Interspeech 2009*. Brighton (UK), 2009.

[7] KUMAR, N. Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition. *Ph.D. dissertation*, John Hopkins University, Baltimore, 1997.

[8] GALES, M. J. F. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions Speech and Audio Processing*, 1999, vol. 7, no. 3, pp. 272-281.

[9] HERMANSKY, H. Perceptual linear predictive (PLP) analysis of speech. *J. Acoust. Soc. Am.*, Apr. 1990, vol. 87, no. 4, pp. 1738 to 1752.

About Authors ...

Jan NOUZA was born in 1957. He received his M.Sc. and Ph.D. degrees at the Czech Technical University (Faculty of Electrical Engineering) in Prague in 1981 and 1986, respectively. Since 1987 he has been teaching and doing research at the Technical University in Liberec. In 1999 he became full professor. His research focuses mainly on speech recognition and voice technology applications (voice-to-text conversion, dictation, broadcast speech processing and design of voice-operated tools for handicapped persons). He is the head of SpeechLab group at the Institute of Information Technology and Electronics.

Jan SILOVSKY (1982) received the Master degree at the Technical University of Liberec (TUL) in 2006. He is currently a PhD student at the Institute of Information Technology and Electronics TUL. His research work is focused on speaker and speech recognition.