Fast Keyword Spotting from Acoustic Baseforms

*Luboš Šmídl, Josef Psutka, Ondřej Obracaník, Petr Podaný, and Jiří Zahradil**

Department of Cybernetics, Faculty of Applied Sciences
University of West Bohemia, Czech Republic
*SpeechTech, s.r.o., Czech Republic
{smidl, psutka, ondrejo, podany}@kky.zcu.cz, jiri.zahradil@speechtech.cz

Abstract

This paper describes a filler model, used in our keyword spotting system, which is implemented as a phoneme recognizer. The filler model produces a sequence of phones corresponding to the input utterance and can be used as a phoneme recognizer. The dependency of accuracy and correctness on the filler model back loop penalty as well as the influence of the filler model language model are depicted. The output of the phoneme recognizer can be used for keyword spotting. Two modifications of basic DTW algorithm are presented. The advantage of this keyword spotting approach is the possibility of two pass detection. The first pass (slow) is done only once. The second pass (fast) is done on the request of searching the keyword and uses only the sequence of the phones generated by the first pass. All the tests are performed on the telephone speech corpus.

1. Introduction

Keyword spotting from the output of the phoneme recognizer is by one order of magnitude faster than the filler model [1] [2], confidence measure [3] or large vocabulary continuous speech recognition (LVCSR) [4] based keyword spotting approaches.

The main advantage of the presented keyword spotting method is the possibility of division into two parts. The first one is processed only once, and it generates a sequence of phones of input utterances. The second one is performed when the request of finding a keyword occurs (see Figure 1).

The second advantage is independence of domain area and independence of speech type (read vs. spontaneous speech). For example, the LVCSR approach cannot be used in a different domain area.

The paper is structured as follows: the phoneme recognizer is presented in Section 2. Section 3 deals with keyword spotting from an output of the phoneme recognizer. The standard detection method and two modifications are introduced. We mainly focus on the processing speed to develop algorithms able to find keyword as fast as possible on the preprocessed data.

The remaining part of the paper presents the experiments as well as results, and concludes this paper.

2. Phoneme recognizer

As a phoneme recognizer we use a filler model from our keyword spotting system described in [1]. The filler model is designed to produce a sequence of phones (acoustic baseform – ABS) corresponding with the input utterance.

The system is speaker–independent and it is based on a statistical approach. It comprises a front–end, an acoustic model, and a decoding block that searches for the best phone sequence matching the acoustic signal.

2.1. Front–end

Speech signal is digitized at 8 kHz sample rate and converted into the mu–law 8–bit resolution format. Then the pre–emphasized acoustic waveform is segmented into 25 milliseconds frames every 10 ms. Hamming window is applied to each frame and static PLP cepstral coefficients (PLP_CCs) are computed. Then delta (first–order derivatives) and delta-delta (second–order derivatives) PLP_CCs are calculated and appended to the static PLP_CCs of the speech frame.

2.2. Acoustic model

A triphone is used as a basic speech unit of the recognition system. Each individual triphone is represented by a 3-state

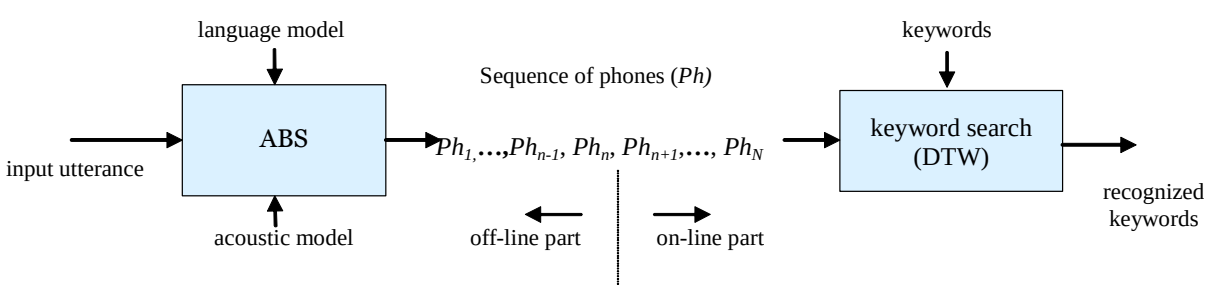

Figure 1: System scheme

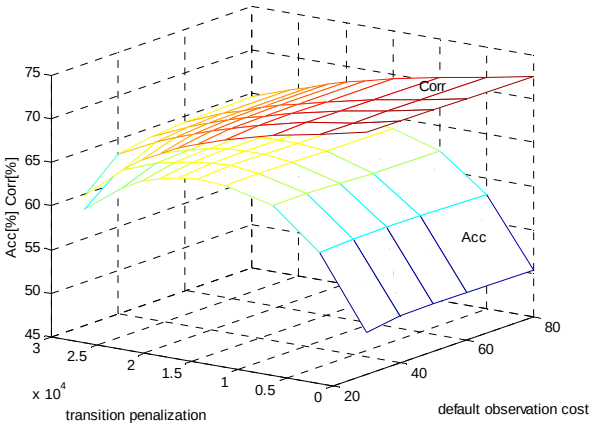

Figure 2: Dependency of accuracy (Acc) and correctness (Corr) on transition penalization

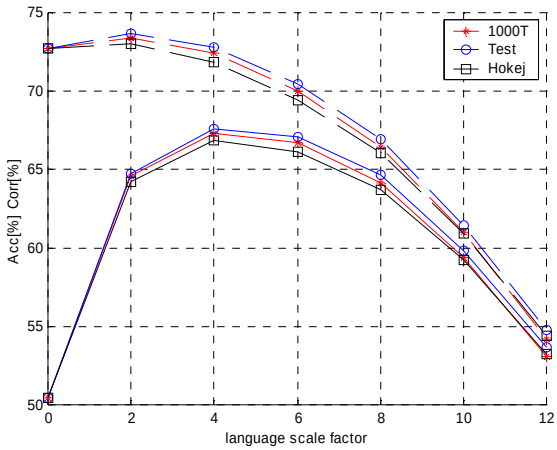

Figure 3: Influence of language model

left–to–right HMM with a continuous output probability density function assigned to each state. Each density is expressed as a mixture of multivariate Gaussians where each Gaussian has a diagonal covariance matrix. The number of mixture components for each state was obtained experimentally. Because a variety of noise sounds, e.g. loud breath, a click on the microphone and the noise of a telephone channel can appear in an utterance, a set of noise HMM models was introduced and trained in order to capture these noise sounds.

2.3. Decoding module

The filler model is decoded via Viterbi search supported by the token passing feature [6]. Within the scope of this paper the term "score of a state" is considered as the cumulative score and denotes the minus-log-likelihood of generating the observation vector sequence given the optimal (in the sense of Viterbi decoding) state sequence. The transition cost and self loop cost are defined as minus-log-likelihood of transition probability and minus-log-likelihood of self loop probability, respectively. The full description of the decoding process is given in [1].

2.4. Filler model

The filler model is constructed as a set of HMM models connected in a parallel way. Each HMM model is composed of a three state left-to-right and represents one context–independent phone or one context–dependent phone respectively. This set is supplemented by silence HMM model and additional non–speech events such as loud breath, coughing, knocking, and noise.

The phone (triphone) transitions in the filler model are penalized to reduce insertion errors. The influence of the phoneme–to–phoneme (triphone–to–triphone) penalization and the language model is shown in Figure 2.

The phoneme bigram language model is used. Three different language models are used – LM trained from:

- 44000 sentences (marked as 1000T) – read speech from an economic area.
- 831 sentences (Test) – read speech from an economic area. These sentences are used for testing the whole system.
- 65 347 sentences (Hokej) – spontaneous speech, transcriptions of championship ice–hockey commentary.

In large vocabulary continuous speech recognition systems, we need enough training data from the same area where the system will be used. In many languages, there is difference of language models for read and spontaneous speech [5].

Because the phones are relatively small units (in comparison with words in LVCSR system) the needs for training data are not so big. The dependency of a quality of LM on a size and area of training data are evident from Figure 2.

3. Keyword spotting from ABS

The method of "moving window" is commonly used for detecting a keyword from an acoustic baseform (ABS) [7]. The beginning of the window in scanned ABS is gradually shifted one char more until the rest of the ABS sequence is at least as long as the minimal value of the desired keyword. In each step, the keyword is compared with *n* (*n=max-min*) parts of the ABS sequence. The minimal (*min*) and the maximal (*max*) length of the selected part should be figured from the length of the desired keyword. For instance, the minimal value is a half of the value of the keyword and the maximum value is one and a half of the value of the keyword. To compute the distance of two words, the DTW algorithm is used. If the distance divided by keyword length is lower than the decision threshold, the keyword is detected. The threshold can be shifted up and down to get a different detection and false alarm rate. While the higher value produces more false alarms and a better detection rate, the smaller value induces more deletions and fewer false alarms.

3.1. Confusion table

The distance of two phonemes is chosen primarily according to prior phonetic knowledge. To improve the match of text transcription and the recognized ABS, a confusion table was implemented into the DTW algorithms. In the confusion table there are stored probabilities of two phoneme substitution. The confusion table was computed from the training corpus marked as 1000T. An example of the confusion table is contained in Figure 4.

3.2. Modification number 1

The method of searching by conducting many computations of DTW in the same beginning time was slightly modified. It is unnecessary to compute the whole table for the desired word and the beginning time *n* times (*n=max-min*). The only DTW table was computed for a word and a part of ABS from beginning time to end time (*=beginning_time + max*).

n	d(*A*(1),*B*(*j*)) +g(1,*j*-1)	d(*A*(i),*B*(*j*)) + | / g(*i*,*j*-1) \ g(*i*-1,*j*-1) \ g(*i*-1,*j*)
1	d(*A*(1),*B*(1))	d(*A*(*i*),*B*(1))+g(*i*-1,1)

B(keyword) → *j*

1 *min* *max*

A (part of ABS) → *i*

The resulting distance of the keyword and part of ABS was as follows:

$$dist(A,B) = \min_{i=min,\ldots,max} g(i, n = length(B)) \quad (1)$$

The distance is then normalized by keyword length and compared with the predefined threshold.

3.3. Modification number 2

The second modification is designed to save computing time as much as possible. This modification slighlty degrades the detection rate but reduces computation demands significantly. For each keyword and input utterance (ABS sequence) the DTW table is computed only once. The standard DTW algorithms have to be modified. There is not only one beginning (position 1,1) for the warping function. The path can start at every position in the first row in the table.

n	d(*A*(1),*B*(*j*)) +g(1,*j*-1)	d(*A*(i),*B*(*j*)) + | / g(*i*,*j*-1) \ g(*i*-1,*j*-1) \ g(*i*-1,*j*)
1	*g*(*i*,1)=d(*A*(i),*B*(1))	

B(keyword) → *j*

1 *min* *Max*

recognized ABS → *i*

If the value in the last row (*n*) divided by the keyword length is lower than the decision threshold, then the keyword is detected at this position (ends at this position). To get the beginning time of the keyword, the back propagation method has to be implemented.

4. Experiments

To evaluate the performance and reliability of the proposed keyword spotting system, the following experiments were carried out. The telephone quality speech corpus (TQSC) was used for all tests. Each speaker uttered 40 sentences. These sentences were uttered by native Czech male and female speakers, and they contain a large number of silent parts and low–level noises. The corpus was randomly divided into three groups. The acoustic models were trained from 1000 speakers. 27 speakers were used for training of the confusion table, and 17 speakers were used for all of the performed tests.

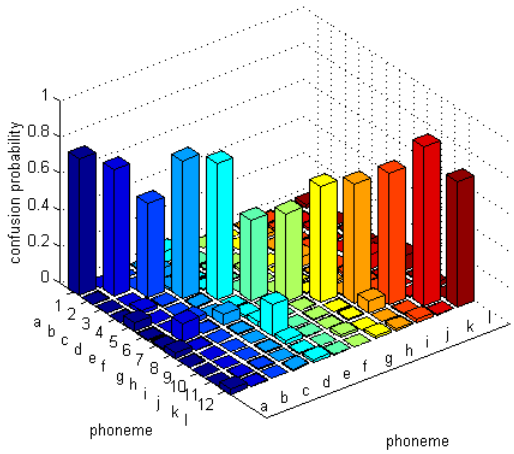

Figure 4: Example of confusion table

4.1. Experiment 1 - ABS

The filler model (phoneme recognizer) is evaluated by phone recognition accuracy (Acc) and correctness (Corr). Table 1 presents the results of phoneme recognition for a monophone model, a triphone model and a triphone model with an implemented bigram language model. The results have been obtained from 831 sentences (17 speakers) from TQSC. The influence of back loop penalization is given in Figure 2. Table 2 shows the influence of the language model. The impact of the quality and domain area of language model is small compared to LVCSR. The results are almost the same for the LM trained from 831 test sentences and from 44000 sentences.

4.2. Experiment 2 – Keyword spotting

The performance of the keyword spotting system was evaluated by the detection rate (DR) and the false alarm rate (FA) defined as follows:

$$DR[\%] = \frac{N_{CORRECT}}{N_{KW}} \times 100 \quad FA[\%/kw] = \frac{FA_{COUNT}}{N_{KW}} \times 100 \quad (2)$$

$$FA\,[1/kw/h] = \frac{FA_{COUNT}}{DURATION_{_TEST} \times KW_{COUNT}} \quad , \quad (3)$$

Table 1: Phoneme recognizer results

	Acc[%]	Corr[%]
monophone	58.09	62.49
triphone	64.29	68.12
triphone + LM	67.33	72.44

Table 2: Impact of phoneme bigram language model

	Acc[%]	Corr[%]
LM: Test	67.56	72.78
LM: 1000T	67.33	72.44
LM: Hokej	66.86	71.83

Table 3: Keyword spotting results

	DR[%] at 10 FA/(kw*h)	**FOM [%]**
baseline	71.2	63.6
modification #1	71.2	63.6
modification #2	70.4	61.4

Table 4: Processing times

	processing time (sec)	**real-time ratio**
off-line part	8013.0	0.49
baseline	1768.3	2.23
modification #1	272.0	14.52
modification #2	22.8	173.20

where $N_{CORRECT}$ and FA_{COUNT} denotes the number of correct detections and false alarms in a spotting result, respectively. N_{KW}, KW_{COUNT}, and $DURATION_{TEST}$ are the total occurrence of the keywords in the tested corpus, number of different keywords, and the total duration of the tested speech corpus in hours, respectively.

The FOM (Figure of Merit) value was also computed. The FOM is defined as the average detection rate from 0 to 10 FA/kw/h (false alarms per keyword per hour).

From 831 test sentences 328 keywords were randomly selected. The total number of keyword occurrences in the test sentences was 382.

All experiments were carried out on the same testing data. The baseline and Modification 1 results are identical. The difference is in the processing speed. Modification 2 slightly decreases the detection rate. The main advantage of Modification 2 is in procesing time. Modification 2 allows us to process input utterances up to 173 times faster than the real–time (Pentium 4 at 3.0 GHz). The times of the processing are presented in Table 4.

5. Conclusions

This paper presents alternative methods for keyword spotting using output of the phoneme recognizer (acoustic baseform ABS). The advantage of this method (and mainly its two modifications) is the processing time. Modification number two is up to 173 times faster than real–time, but it slightly decreases the detection rate (from FOM 63.6% to 61.4%). In addition, the phoneme recognizer is evaluated. The influence of the transition penalty is presented. The phoneme language model is discussed. As has been shown, there is no need for a large training set, nor for the demand of the same domain area (topic and read vs. spontaneous speech), in contrast to the LVCSR word based language model.

The future work is to modify the phoneme recognizer to produce a phoneme lattice instead of phoneme sequence and then to search for a keyword in this lattice. This will results in increasing phoneme recognition accuracy, together with the increasing keyword detection rate.

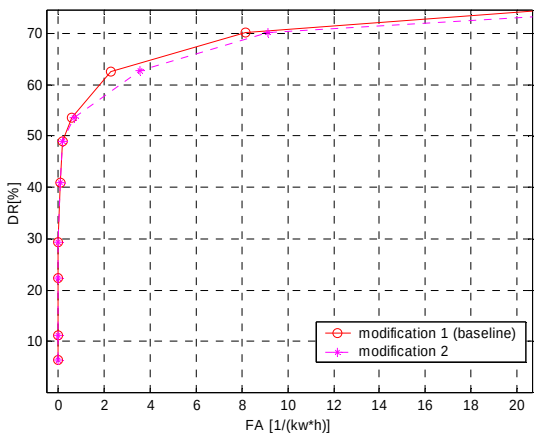

Figure 5: ROC characteristic

6. Acknowledgements

This work was supported by the Grant Agency of Academy of Science of the Czech Republic, project no.1QS101470516.

7. References

[1] Šmídl L., Müller L., "Keyword Spotting for Highly Inflectional Languages", *The Proceedings of the 8th International Conference on Spoken Language Processing, ICSLP 2004 (INTERSPEECH), Jeju, Korea, pp. 297-300,*

[2] Manos, A. S., Zue, V. W., "A segment-based wordspotter using phonetic filler models", *Proc. ICASSP'97, pp. 899—902*

[3] Junkawitsch, J., Neubauer L., Höge H., and Ruske G., "A new keyword spotting algorithm with pre-calculated optimal thresholds", *International Conference on Spoken Language Processing ICSLP 1996, Philadelphia, USA, 1996, pp. 2067-2070*

[4] Weintraub, M., "Keyword-spotting using SRI's DECIPHER large-vocabulary speech recognition system", *Proc. EUROSPEECH'93, pp. 1265-1268*

[5] Psutka, J., Ircing P., Hajic, J., Radova, V.,Psutka, J.V., Byrne, W., and Gustman, S., "Issues in annotation of the Czech spontaneous speech corpus in the MALACH project", *Proceedings of the International Conference on Language Resources and Evaluation*, LREC, 2004.

[6] Young, S. J., Russell, N. H., Thornton, J. H. S., "Token Passing: a Simple Conceptual Model for Connected Speech Recognition Systems", *Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Department, July 31, 1989.*

[7] Vopička, J., "Vyhledávání klíčových elementů v souvislé promluvě", (in Czech), *Ph.D. Thesis*, ČVUT Praha, 2002.