Comparison of Keyword Spotting Methods for Searching in Speech

Luboš Šmídl and Josef V. Psutka

Department of Cybernetics,
Faculty of Applied Sciences,
University of West Bohemia, Czech Republic
smidl@kky.zcu.cz, psutka_j@kky.zcu.cz

Abstract

This paper presents and discusses keyword spotting methods for searching in speech. In contrast with searching in text, the searching in speech or generally in multimedia data still represents a challenge. The aim of the paper is to present a keyword spotting (KWS) method based on a large vocabulary continuous speech recognition (LVCSR) system, based on phonetics decoder, and keyword spotting using a filler model. All the methods are evaluated and compared from various points of view – speed, quality, requirements on training data and so on. All experiments are done using a telephone-quality speech corpus. Furthermore, this paper presents a new block decision in filler model-based keyword spotting which brings the speedup of decision together with better detection.

Index Terms: keyword spotting, searching in speech, speech recognition, LVCSR, filler model, acoustic baseform

1. Introduction

Keyword spotting is a special branch of automatic speech recognition dealing with detecting a limited number of words in an utterance. This paper discusses a statistical and HMM-based approach to keyword spotting.

The problem of detecting a limited number of keywords can be solved in three major ways. These approaches are presented here, and their advantages and disadvantages are considered. The most obvious approach is to use a large vocabulary continuous speech recognition system to produce a word string, and then to search for the keyword in this word string. Theoretically, this is the best way, but there are problems with out-of-vocabulary words, false starts, hesitations, repetition, and other irregularities. The second presented approach is based on analyzing the output of the phonetics decoder – acoustic baseform (ABS).

The third, new approach, which combines the filler model with the confidence measure approach, is presented here. We evaluate the filler model score to obtain a normalization factor simultaneously with a keyword score. This normalization makes the algorithm independent of the keyword's phoneme composition. In order to decide if a keyword was or was not spoken, a normalized score of the keyword (*KWNS*) is compared with a predefined threshold. Furthermore, the block decision is introduced. With this approach, the decision is not carried out in each decoding time frame, but only after the token stored in the last keyword state is changed. This approach allows more information to be used in a decision; in addition to *KWNS*, other values, such as the length of the keyword, can be considered in the decision.

2. System Overview

The system is speaker-independent and is based on a statistical approach. It comprises a front-end, an acoustic model, and a decoding block. The front-end and acoustic model are the same for all experiments, like training/testing data.

2.1. Front-end

The speech signal is digitized at 8 kHz sample rate and converted to the mu-law 8bit resolution format. Then the pre-emphasized acoustic waveform is segmented into 25 milliseconds frames every 10 ms. A 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

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

Since a variety of noise sounds, e.g. loud breath, click on the microphone and 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

In the scope of this paper the term "score of a state s of a HMM model m in time t" is considered as the cumulative score, denoting the minus-log-likelihood of generating the beginning of the observation vector sequence up to the time t given the optimal (in the sense of Viterbi decoding) state sequence which ends at time t in state s. For example, if the probability density $p(o_1, o_2, \ldots, o_t \mid q_{1m}, q_{2m}, \ldots, q_{tm})$ that the sequence of observation vectors $o_1, o_2, \ldots, o_t$ is generated by the HMM state sequence $q_{1m}, q_{2m}, \ldots, q_{tm}$ of the HMM model M, then the score $s(q_{tm}, t)$ of the state q_{tm} in time t is $-\log(p(o_1, o_2, \ldots, o_t \mid q_{1m}, q_{2m}, \ldots, q_{tm}))$. 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.

2.4. Training/testing data

To evaluate the performance and reliability of the proposed keyword spotting systems, the following experiments were provided. The telephone speech corpus (TQSC) was used. Each speaker uttered at least 40 sentences. These sentences were spoken by native Czech male and female speakers, and contain a large number of silence parts and noises. The corpus (1050 speakers) was divided into three groups. The acoustic models were trained from 1000 speakers. 33 speakers were used for training the decision module, and 16 speakers were used for all tests.

From 832 test sentences (duration 3948.94 sec) containing 3446 different words 328 keywords were selected with the following limitation: the minimal length of a keyword was three phones, and the selected keywords had to differ in more than two phones from each other. The total number of occurrences of keywords in the test sentences was 381.

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

$$DR\ [\%] = \frac{N_{CORRECT}}{N_{KW}} \times 100 \quad FA\ [\%/kw] = \frac{FA_{COUNT}}{N_{KW}} \times 100 \qquad (1)$$

$$FA\ [1/kw/h] = \frac{FA_{COUNT}}{DURATION\_TEST \times KW_{COUNT}} \quad , \qquad (2)$$

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, the 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). The equal error rate (*EER*) was computed to make results comparable. The *EER* is defined as an intersection point between the false alarm curve and the $1 - DR$ curve (false rejection curve).

3. Keyword spotting

3.1. Filler model approach

We present a novel method which takes advantage of the filler model and confidence measure based keyword spotting [1]. We evaluate the filler model score to obtain a normalization factor simultaneously with a keyword score. This makes our decision algorithm independent of the keyword phoneme composition. The scheme of the decoding block is in Figure 1.

All keywords are represented by concatenation of their triphone models, with full left and right contexts. The filler model is constructed as a set of HMM models connected in a parallel way. The phoneme bigram language model is used. The influence of the phoneme language model is discussed in [2].

The filler model is placed in front of the keyword models, so the start state of each keyword is linked with the output of the filler (see Figure 1). If the filler to keyword (*F2K*) transition is performed, then the transition time $t = t_{F2K}$, the best score of the

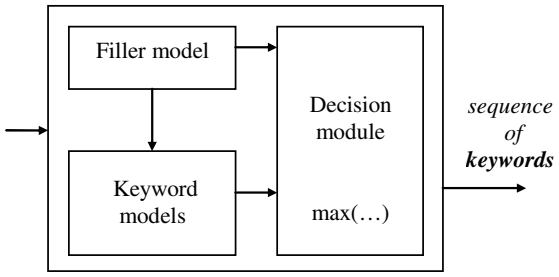


Figure 1. *Decoding block architecture.*

third states of the filler model $FMBestScore(t_{F2K})$, and the score $KWs_1(i, t_{F2K})$ are stored in the first state of the keyword *i*. During the Viterbi decoding the stored values are propagated through the keyword model *i* until they are replaced with a token which has a better (i.e. lower) score $KWs_k(i, t' > t_{F2K})$ or reach the last state of the keyword *i*.

The filler model and keyword models are decoded via Viterbi search supported by the token passing feature [3]. The full description of the decoding process is given in [1].

The aim of the decision module is to reject or accept a keyword hypothesis if a given keyword *i* ends in the time *t*. Two methods of decision were implemented. The first one, frame–to–frame (F2F) decision, was introduced in [1].

The second one, block decision (BD), is carried out only in the time (changing time) when the record ($t_{FTK}(s_{last}(i,t))$ and $FMBestScore(t_{F2K}(s_{last}(i,t)))$) stored in the last keyword state is changed. Between two changing times there is a history of evolution of the last keyword state score and a history of evolution of the best filler model state score. From both histories there are used for the decision these values: length of the phonetic transcription of the keyword, minimal value of smoothed keyword normalized score $SKWNS(i,t)$, and time of this minimal value. The $SKWNS(i,t)$ is defined as follows:

$$SKWNS(i,t) = \frac{SKWNS(i,t-1) + KWNS(i,t)}{2} \qquad (3)$$

$$KWNS(i,t) = \frac{KWs_{last}(i,t) - FMBestScore(t_{F2K}(s_{last}(i,t)))}{FMBestScore(t) - FMBestScore(t_{F2K}(s_{last}(i,t)))} \qquad (4)$$

These values form a feature vector $\boldsymbol{x}$ for each changing time. This vector is classified by a linear discriminative hyperplane ($g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$) into two classes: keyword occurrence hypothesis acceptation × rejection. The distance from the discriminative hyperplane is used as a certainty factor of

Table 1. *Filler model approach keyword spotting results.*

method	**F2F**	**BD**
EER [%]	22.31	15.58
FOM [%]	93.61	97.80
DR [%] at 0.1 *FA/h/kw*	73.75	81.78
DR [%] at 10 *FA/h/kw*	96.23	99.48

keyword detection. The hyperplane can be shifted up or down by threshold value to get different detection rate and false alarms rate. The results are presented in Table 1.

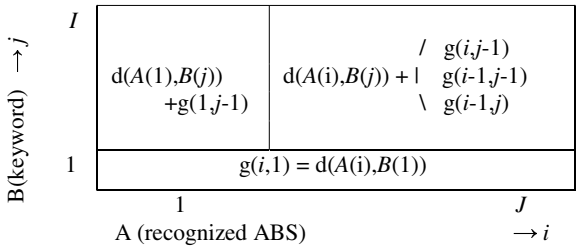
3.2. Phoneme recognition approach

As a phoneme recognizer we use a filler model from our keyword spotting system described in Section 3.1. The filler model is designed to produce a sequence of phones (acoustic baseforms – ABS).

The method of "moving window" is commonly used for detecting a keyword from an acoustic baseform (ABS) [4]. 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 distance of two phonemes d(A,B) 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.

To speed-up the algorithm the basic method was modified. For each keyword and input utterance (ABS sequence) the DTW function 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, but the path can start at every position in the first row in the table (ABS sequence).



If the value in the last row (I) divided by the keyword length (I) 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 was implemented.

The filler model (phoneme recognizer) is evaluated by phone recognition accuracy (*Acc*) and correctness (*Corr*). Table 2 presents the results of phoneme recognition for a triphone model a with an implemented bigram language model.

Table 2. *Phoneme recognizer results.*

Acc [%]	*Corr* [%]
67.33	72.44

Table 3. *Keyword spotting from ABS.*

FOM [%]	61.41
EER [%]	47.45
DR [%] at 0.1 *FA/h/kw*	40.84
DR [%] at 10 *FA/h/kw*	70.16

The results of keyword spotting are presented in Table 3.

3.3. LVCSR approach

Theoretically, the best way for keyword spotting is LVCSR based approach, because besides an acoustics model we use a language model as well. The second advantage is that this approach is easy to implement. The LVCSR system produces a sequence or lattice of words, and then we only search for a keyword in this sequence/lattice.

One known problem arises in out-of-vocabulary words (mainly unique names of people, companies, places, …), false starts, hesitation, repetition, etc. The second problem is presented by the potential preparation of an appropriate language model. In many languages (especially Slavic languages) the language model for read speech is different from the language model of spontaneous speech [5]. The read speech language model can be build from large amount text data for example downloaded from newspapers web sites. But the spontaneous language model has to be built from manually done transcription of conversations, public debates and discussion.

To analyze the influence of the type of the language model, the following experiments were performed. We used three language models:

- LM1 – ideal language model, built from all training and testing utterances. This LM was used to find out theoretically the best results,
- LM2 – language model from a different area than testing data (survivors of the Holocaust) and different type of speech (spontaneous, very emotional) – to test the theoretically worst results,
- and LM3 – real language model, the same domain area (economic news) and the same type of speech (read speech), not containing train and test sentences.

The results (accuracy (Acc) and correctness (Corr)) for all language models together with test data perplexity (ppl) and OOV are presented in Table 4. For all performed keyword spotting experiments only the best sequence of words was used as output of the LVCSR system. The keyword spotting results are given in Table 5.

Table 4. *Filler model approach keyword spotting results.*

language model	**LM1**	**LM2**	**LM3**
words	39441	41687	40109
ppl	32.74	3717.17	699.16
OOV [%]	0	17.19	7.77
Acc [%]	83.96	50.25	55.00
Corr [%]	89.44	51.63	67.31

Table 5. *LVCSR based keyword spotting results.*

language model	**LM1**	**LM2**	**LM3**
DR [%]	88.19	55.38	69.82
FA [1/h/kw]	0	0.028	0.028

Table 6. *Real time ratio.*

keyword spotting method	**FM**	**ABS**	**LVCSR**
processing time [s]	2705.5	22.8	18533.5
RT ratio	1.46	173.20	0.215

4. Conclusions

This paper presents new keyword spotting system taking advantage of both the filler model and the confidence measure based approaches. The block decision results in better detection simultaneously with the speedup of the decision. The system is able to operate in real-time. The FOM is 97.8 %. The main advantage is the universality of this method; we can use it in different domain areas and for other languages without tuning the system. The second advantage is that we can simply shift the decision threshold to obtain a different detection rate and false alarm rate.

The results were compared with two methods. The first method is an alternative method for keyword spotting using output of the phoneme recognizer (ABS). The advantage of this 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. Processing a sequence of phones is up to 173 times faster than real–time (for 381 keywords). Let us mention that for one keyword this method is up to 750 times faster than RT. Unfortunately, the FOM is only around 60 %.

The third method used for keyword spotting is LVCSR based approach. By using an appropriate language model, the false alarm rate is lower by one order of magnitude at specific detection rate while the time of processing is the highest of all the presented methods. Without a language model, the LVCSR approach cannot reach as high a detection rate as the presented methods. The LVCSR approach works almost without false alarms, and is the best, if we do not require as high as possible detection rate.

All the methods were compared to each other from the point of view of speed in Table 6. The processing time and the RT ratio are for all keyword (381). The ROC characteristics are presented in Figure 2. For the LVCSR approach there is only one value (we cannot set the system to get a different detection rate and a false alarm rate). For each language model the results are highlighted by a dashed line at the same level.

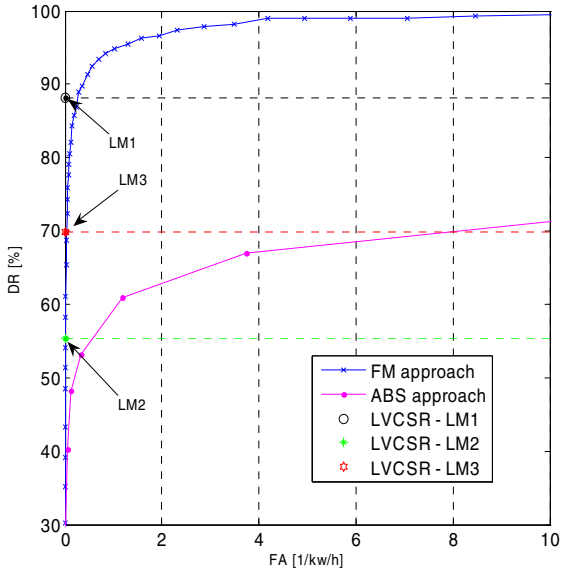


Figure 2. *ROC characteristic.*

5. Acknowledgements

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

6. 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, 2004.

[2] Šmídl, L., Psutka, J., Zahradil, J., "Keyword spotting with triphone based filler model". In SPECOM 2005, Moscow State Linguistics University, pp. 487-490, 2005.

[3] 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.

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

[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.