Improving a Keyword Spotting System Using Phoneme Sequence Generated by a Filler Model

LUBOŠ ŠMÍDL 1, LUDĚK MÜLLER 2, and JOSEF PSUTKA 2

1 Center of Applied Cybernetics, 2 Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, CZECH REPUBLIC

Abstract: - This paper presents a technique for improving keyword spotting. The main idea is based on the combination of a standard keyword spotting system working with a filler model with a phone recognizer. If the filler model is well designed it produces a phoneme sequence corresponding to the investigated utterance. This phoneme sequence generated by a filler model can be used to reject incorrectly detected keywords. The quality of this technique depends on how good is the acoustic match between an incoming utterance and a bigram phoneme filler model.

Key-Words: - keyword spotting, filler model, confidence measure, acoustic baseforms.

1 Introduction

Keyword spotting can be used in many applications such as information retrieval, topic detection, dialog systems with simple understanding based on keywords, finding unique names of places or peoples in large spoken archives, and monitoring radio and/or television broadcasting. The problem of detecting a limited number of keywords can be solved in three major ways.

The most obvious way is to use a large vocabulary continuous speech recognition system (LVCSR) to produce a word string, and then search for keyword in this word string [1]. Theoretically this is the best way, but there are problems with out-of-vocabulary words, false starts, hesitations, repetition and etc. Latest research experiences show that it is nowadays technically impossible to make a recognizer with a vocabulary that covers all words uttered during practical application.

The second way is a use of filler (garbage, mumble) model which represents the non-keyword parts of the utterance [2]. In this method the recognition is made with continuous speech recognition systems (CSR) using a grammar with parallel connection of the filler model and keyword models. The disadvantage of this approach is that one false detected keyword can affect another keyword leading to a large number of indirect causes for misrecognition [3].

The third approach is based on a confidence measure which gives us directly an estimate of a-posteriori probability of occurrence the keyword at a given position in the utterance. We use a derivat of this approach. We evaluate the filler model score to obtain a normalization factor simultaneously with a keyword score. This normalization makes our algorithm independent on keyword phoneme composition. In order to decide if a keyword was or was not spoken a normalized score of the keyword is compared with a decision threshold.

The paper is structured as follows: In Section 2 acoustic modeling with log-likelihood scores normalization is depicted, the keyword spotting system architecture and the decoding process based on the normalization of keyword models by related filler model score is presented. The Section 3 presents a technique of improving a keyword spotting system based on comparing the phoneme sequence generated by a filler model with phonetic transcription of the keyword. In Section 4 the experiments and the results are given. Finally, the last Section 5 concludes this paper.

2 System Description

The keyword spotting algorithm is incorporated in a telephone dialog system designed at the Department of Cybernetics at the University of West Bohemia in
Pilsen [4]. This system is used in many practical applications, for example “VoiceXML Based Telephone Dialog System Providing Access to Entrance Examination Results Stored in the University Database” [5]. Using keyword spotting technology gives the system a chance to discover the keyword command in the whole sentence.

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

2.1 Acoustic Modeling

As a basic speech unit of our recognition system a triphone is used. A triphone is a phone situated between two specific phones. Each individual triphone is represented by a tree-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 mixtures for each model was obtained experimentally.

Because 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. Modeling of pieces of utterances such as the loud breath could be a good example of this process.

The speech signal is digitized at 8 kHz sample rate and converted to the mu–law 8–bit resolution format. The parameterization process is based on the Perceptually-based Linear Predictive (PLP) analysis. Firstly the pre–emphasized acoustic waveform is segmented into 25 milliseconds frames every 10 ms. Hamming window is applied to each frame and 7 (static) PLP cepstral coefficient (PLP_CCs) are computed. Then 7 delta (first–order derivatives) and 7 delta-delta (second–order derivatives) PLP_CCs are calculated and appended to the static PLP_CCs of each speech frame.

The tied states are represented in the space of dimension 21 (7 PLP_CCs + 7 delta + 7 delta-delta) by the mixture of 8 Gaussian distributions in our telephone dialogue system. During the decoding it is necessary to compute a large number of minus log-likelihood scores (LLSs) for the input vector of each time frame (every 10 ms). To implement a real time speech recognizer with a small computation cost it is important to reduce or simplify the large number of calculations.

The solution of this problem is a technique establishing relatively exactly (in an original space of dimension 21) the first 150 best (most probable) LLSs. This technique uses relevant statistical properties of the Gaussian mixture densities combining them with a quick selection and evaluation of a group of the most probable Gaussians together with the k-NN method. Such approach allows more than 90% reduction of the computation cost without any substantial decrease of recognition accuracy.

The basic unit for keyword spotting is a context–independent phone. The triphone-based models used in continuous speech recognition system have to be converted for this reason to context–independent phone. Actually the probabilities of emission of an observation vector in a given state are evaluated as the maximal emission probability of all corresponding states of context–dependent triphones. Thus neither additional HMM models nor additional training is required.

The LLSs of observation vectors have a large dispersion. For this reason we used a normalization that gives the same value for the best and the same value for the worst log-likelihood score in each time. The best score (in each time) is subtracted from the each log-likelihood score and the result is normalized by factor computed from the best and the worst score. After this normalization all LLSs are in a predefined range. It is important to know the best and the worst LLSs values for setting other parameters in the algorithm.

2.2 Keyword spotting

As it was mentioned above there are two main approaches usable in practical applications. In the approach that uses a filler (mumble, garbage) model a Viterbi search finds out the best path through the recognition network. For any non–keyword part of an utterance the filler model should have a better acoustic match than any keyword model; thus insignificant words are assigned to the non–keyword part of the utterance. In this way the filler model catches the non–keyword parts. Finally, the “filler” words are omitted from the resulting recognized word sequence and only the key–words remain on the output. During decoding one keyword can affect another (the filler model competes with the keyword model). For this reason a problem with a large number of indirect causes for misrecognition arises.

In the approach without a filler model it is necessary to use a modification of the Viterbi search algorithm because the keyword can start at any
position in the utterance. The solution is based on a comparison of a keyword model score with a threshold which is generally depended on the keyword.

We present a keyword spotting system that combines advantages of the both approaches.

2.2.1 KWS System Architecture
The presented keywords spotting system is able to indicate and classify predefined words within continuous speech. There are no restrictions in keyword concurrencies – each utterance may contain any number of keywords.

A keyword is simply represented by concatenation of phone models, so no special training data is needed to model keyword. If the keyword has more than one phonetic transcription each of them is represented by its HMM model. For the initialization of keyword models a filler model is used. The filler model is shown in Figure 1 and is constructed as a set of HMM models connected in a parallel fashion. Each HMM model is a three-state left-to-right and represents a context-independent phone. This set is supplemented by HMM model of silence and additional non-speech events such as loud breath, coughing, knocking, and noise. Phoneme bigram is used as the language model. The connection of keyword models and the filler model is shown in Figure 1.

2.2.2 Decoding
At the beginning of decoding the filler model starts with a zero score for each state. In this way the keyword spotting system is initialized. For going through the filler model and the keyword models a standard Viterbi search technique is used. The phone transitions in filler model are penalized to ensure a better acoustic match of the investigated utterance. The filler model generates the best phone sequence. In a common keyword spotting system the filler model is used only for the absorption of a non-keyword part of the utterance. As it will be mentioned below, in our system the filler model is used for normalization of keyword acoustic score.

The keyword models are initialized by the best score of the filler model final state. If the first states score (LLS) of the keyword model is higher than the best score of the filler model final state (plus the transition penalty from the filler to the keyword) the filler to keyword (FTK) transition is done. By this transition the current time and the KW incoming score (it is equal to the best score in the filler model final state in the current time) is stored in the state. This stored information is propagated during Viterbi decoding (as token passing [6]) together with the best score to the last state of the keyword.

For the decision if the keyword was spoken or was not spoken the following information is used: an incoming score of the keyword (\(KWScore_{IN}\)), an incoming time of the keyword, a score of the last keyword state (\(KWScore_{OUT}\)), and the best score of the filler model at time of evaluating the last keyword state (\(FMBestScore\)).

Many research teams use for the decision of acceptation/rejection the score of the last keyword state minus the incoming keyword score divided by the keyword length in frames or states. Unfortunately, this measure depends on keyword phones and length. The decision threshold is shifted based on a keyword structure. To avoid the threshold tuning we present a new measure technique defined as follows:

\[
CM = \frac{KWScore_{OUT} - KWScore_{IN}}{FMBestScore - KWScore_{IN}}
\]
where $FMBestScore$ is the best score of the filler model in time of evaluating the last keyword state. Let us note that $KWScore_{KWScore}$ is equal to the best score of the filler model at time when the FTK transition was done. The measure normalization plays a very important role. If the incoming utterance matches the keyword model, the score of the last state of keyword model falls down and attacks the level of the filler model best score.

The decision if the keyword was spoken or was not spoken is based on comparison of the normalized score of the last keyword state with a predefined threshold. The threshold can be shifted up or down to get different detection and false alarm rate. While the higher value produces more false alarms and better detection rate, the smaller value induces more deletions and few false alarms.

### 3 Using Phoneme Sequence

If the filler model is well designed as phoneme model then it is able to provide a phoneme sequence corresponding to the investigated utterance. This phoneme sequence could by used as a support for the decision algorithm. The phoneme sequence from the keyword start time to the keyword end time should be similar to the phonetic transcription of the investigated keyword. For each keyword candidate (the keywords score is smaller than a guessing threshold) the phoneme sequence generated by the filler model and corresponding to the keyword boundaries is compared with a keyword phonetic transcription.

The phoneme distance is determined by DTW matching. To support a language independency a simple local phoneme distance is used: the zero penalties if the compared phones are equal otherwise 1. The final DTW distance is normalized by the length of phonetic transcription. After this the distance does not depend on the keyword length. The distance assumes values between zero (filler phoneme sequence equals the phonetic transcription of the keyword) and 1 (there are no correspondence between the phonetic keyword transcription and the filler phoneme sequence).

The phoneme similarity score is added to the original score and the result is compared with the decision threshold. The illustration of this process is presented in Figure 2.

### 4 Experiments

To evaluate the performance and reliability of our keyword spotting system, the following experiments were provided. For the tests 44000 utterances (sentences) from the telephone speech corpus [7] were used. Each speaker recorded 40 sentences. These sentences were spoken by native male and female speakers and contain a lot of silence and noise parts. The corpus was randomly divided. 100 speakers were selected for testing. From the rest (1000 speakers x 40 sentences) the acoustic model was trained.

From the 43 test sentences 162 keywords were randomly selected with the following limitation: the minimal length of the keyword was four phones, and the selected keywords have to differ in more then one phone to each other. The total occurrence of keywords in testing sentences was 181.

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

$$DR[\%] = \frac{N_{\text{Correct}}}{N_{\text{Keyword}}} \times 100$$

$$FA[1/kw/h] = \frac{FACOUNT}{DURATION_{\text{Test}} \times KW_{\text{COUNT}}}$$
The value $N_{\text{CORRECT}}$ and $F_{\text{ACOUNT}}$ is the number of correct detections, and false alarms in a spotting result. $N_{\text{KEYWORD}}$, $KW_{\text{COUNT}}$, and $DURATION_{\text{TEST}}$ is the total occurrence of the keywords in the tested corpus, number of kinds of keywords, and the total duration of the tested speech corpus in hours, respectively.

The value FOM was computed to compare the baseline system with presented modification. The FOM – Figure of Merit [8] is defined as the average detection rate from 0 to 10 FA/kw/h (false alarms per keyword per hour).

The FOM value for baseline system is 81.4% and the detection rate at a false alarm rate of 10 fa/kw/h is 86.9 %. Using the phoneme sequence the detection rate at a false alarm rate of 10 fa/kw/h is 87.8 % and the FOM is 83.1 %. Figure 4 shows the dependency of the detection rate on number of false alarms (per keyword per hour).

5 Conclusions
This paper describes a technique for keyword spotting system improvement. The technique uses a property of well designed filler model. A necessary condition is that the filler model should represent investigated utterance sufficiently. Then the phoneme sequence generated by the filler model could by used as a support for rejection/acceptance decision.

7 Acknowledgment
This work was supported by the Ministry of Education of the Czech Republic under project LN00B096.