# EFFICIENT METHODS FOR DETECTING KEYWORDS IN CONTINUOUS SPEECH

*Jochen Junkawitsch*[1], *Günther Ruske*[2], *Harald Höge*[1]

[1]Siemens AG, Otto-Hahn-Ring 6, D-81730 Munich, Germany
[2]Institute for Human-Machine-Communication, Munich University of Technology, Germany

email: Jochen.Junkawitsch@mchp.siemens.de

## ABSTRACT

This paper refers to our prosperous development of algorithms for detecting keywords in continuous speech. Two different approaches to define confidence measures are introduced. As an advantage, these definitions are theoretically calculable without artful tuning. Moreover, two distinct decoding algorithms are presented, that incorporate these confidence measures into the search procedure. One is a new possibility of detecting keywords in continuous speech, using the standard Viterbi algorithm without modeling the non-keyword parts of the utterance. The other one is an improved further development of an algorithm described in [1], also without the need of modeling the non-keyword parts.

## 1. INTRODUCTION

The problem of detecting a limited number of keywords in continuous speech can be solved in two major ways. The first possibility is the application of a large vocabulary speech recognizer ([2]). But this approach is very extensive and has some difficulties in dealing with out-of-vocabulary words and out-of-grammar sentences. Evading this trouble and resulting in a much more simple solution, the second way is to build up a word spotter using only the keyword models in connection with specific garbage or filler models representing the non-keyword parts of the utterance ([3]). But this implies the problem of providing a qualified and accurate realization of general filler models.

So in both cases the common problem is the manner of dealing with the out-of-vocabulary parts of the speech signal. In this paper, an important difference in modeling non-keyword speech is pointed out:

- Modeling non-keyword speech beyond the word boundaries of an assumed keyword (i. e. a filler model in the true sense of the word).
- Estimating the probability of an out-of-vocabulary word when a certain keyword $W$ is supposed by the recognizer (i. e. modeling $P(\overline{W}|O)$ during a keyword is uttered).

All approaches to keyword detection introduced in this paper renounce modeling non-keyword speech outside the keyword boundaries and are restricted to an estimation of $P(\overline{W}|O)$ during a keyword utterance.

The aim of this paper is to present a search strategy for keyword spotting, which is based on the maximization of a confidence measure. Therefore, two possible definitions of a confidence measure are introduced, that can be used for keyword detection. These definitions are based on the estimated value of $P(\overline{W}|O)$ solely within the keyword boundaries and can be theoretically deducted and calculated only using one general target HMM set. Moreover, two different algorithms are proposed and compared, that maximize these confidence measures without the need of garbage or filler models. The first one is a variation of the commonly used Viterbi algorithm, whereas the second one is an improved and further developed version of a decoding algorithm based on normalized scores ([1]).

As an important advantage of these methods, no representation of non-keyword speech outside the keyword boundaries is necessary. Therefore unknown words and noises, that occur before or after a keyword, don't disturb the detection algorithm. Moreover, no language model is used within the search, that means no grammar is restricting keyword occurrences even in ill-formed sentences. Finally, the calculable definition of the confidence measures solves the problem of discriminatively trained HMM sets and the delicate subject of tuning filler models.

## 2. WORD SPOTTING ALGORITHM

### 2.1. Search strategy

The starting point in speech recognition theory can be expressed as the goal of maximizing the a-posteriori probability of a word sequence. Usually this is noted as follows:

$$W = \underset{\omega}{\operatorname{argmax}}\ P(\omega|O) = \underset{\omega}{\operatorname{argmax}}\ \frac{P(O|\omega) \cdot P(\omega)}{P(O)}$$

In the case of keyword spotting, this strategy has some deficiencies. When the estimation of the likelihood $P(O|\omega)$ is done by utilizing HMMs, every part of the speech signal has to be represented by a complying model, but only a few keywords are interesting. Moreover, this approach doesn't imply any rejection technique. Finally, it has been already noticed in former papers, that in order to improve recognition performance it is profitable to integrate a confidence measure into the search process ([4]). Therefore, in the case of keyword spotting it may be prosperous to change this basic search strategy in a way, that a suitable rejection mechanism and the application of a confidence measure are included and no other parts of the speech signal than the keywords themselves have to be modeled by HMMs. For this purpose an approach based on a confidence measure $C$ is examined:

$$C(W|O) > \textit{threshold} \Rightarrow W \text{ accepted}$$

This method can be interpreted as optimizing a certain confidence measure for each single keyword separately within the decoder and comparing it with a definite threshold that can be chosen individually for each keyword.

## 2.2. Confidence measures

Two specific possibilities for defining confidence measures are introduced and examined. These confidence measures serve as an optimization criterion within the search procedure as well as they are used in order to rate keyword hypotheses and to decide between keyword rejection and acceptance.

The **first definition $C_1$** simply declares the negative logarithm of the keywords a-posteriori probability as the confidence measure:

$$C_1 = -\log P(W|O)$$

This approach is consistent to the principle of optimizing $P(W|O)$, which is generally used within speech recognition systems. Therefore it is the most obvious definition of a confidence measure. In order to pass over to the frame level, the Bayes' rule is applied in conjunction with the following assumptions:

$$P(O) = \prod_t P(O_t)$$

$$P(W) = \prod_t P(s_{\psi(t)})$$

$$P(O|W) = \prod_t [P(O_t|s_{\psi(t)}) \cdot a_{\psi(t-1),\,\psi(t)}]$$

The probability of a sequence of feature vectors $P(O)$ is expressed by the multiplication of the probabilities of the single feature vectors $P(O_t)$. In the same way the probability $P(W)$ of a whole word is calculated by multiplying the single probabilities $P(s_{\psi(t)})$ of each selected state of the HMM-set, where $\psi(t)$ is a function, which maps the time variable $t$ to the state number chosen by the decoder. The likelihood $P(O|W)$ is considered to be the usual HMM likelihood, which can be computed using the emission probabilities $P(O_t|s_{\psi(t)})$ and the transition probabilities $a_{\psi(t\text{-}1),\psi(t)}$. In this way the confidence measure $C_1$ can be noted as:

$$C_1 = \sum_t -\log\left(\frac{P(O_t|s_{\psi(t)}) \cdot P(s_{\psi(t)})}{P(O_t)} \cdot a_{\psi(t-1),\,\psi(t)}\right)$$

Considering the working method of the Viterbi algorithm, this equation suggests the definition of a local confidence score $c_1(O_t|s_j)$, that can be used within the search procedure:

$$c_1(O_t|s_j) = -\log\frac{P(O_t|s_j) \cdot P(s_j)}{P(O_t)}$$

The probability of a feature vector, which appears as the denominator, can be calculated by taking all states of the HMM-set into account:

$$P(O_t) = \sum_k P(O_t|s_k) \cdot P(s_k)$$

The a-priori probabilities $P(s_k)$ of the states can be determined in advance within the training procedure, thus the local confidence score $c_1(O_t|s_j)$ is completely calculable.

Furthermore, a **second definition $C_2$** of a confidence measure is realized by a likelihood ratio, consisting of the conditional probabilities of a certain feature vector sequence $O$ given a particular keyword model $W$ and given an assumed and corresponding anti-model $\overline{W}$ respectively:

$$C_2 = -\log\frac{P(O|W)}{P(O|\overline{W})}$$

The anti-model $\overline{W}$ doesn't really exist, but its emission probability can be calculated. In contrary to the first definition, this method leads to a symmetrical confidence measure with a balance value zero, if $P(O|W) = P(O|\overline{W})$ is fulfilled. Again, the transition to frame level variables is performed in a similar way and (using $\overline{a_{\psi(t-1),\,\psi(t)}}$ as transition probabilities within the assumed anti-model$\overline{W}$ consisting of anti-states $\overline{s_{\psi(t)}}$) results in the following equation:

$$C_2 = \sum_t -\log\frac{P(O_t|s_{\psi(t)}) \cdot a_{\psi(t-1),\,\psi(t)}}{P(O_t|\overline{s_{\psi(t)}}) \cdot \overline{a_{\psi(t-1),\,\psi(t)}}}$$

A suitable local confidence score $c_2(O_t)$, which can be applied within the search, can be defined by:

$$c_2(O_t) = -\log\frac{P(O_t|s_j)}{P(O_t|\overline{s_j})}$$

In this case the local confidence score $c_2(O_t|s_j)$ is calculable, too, because the denominator can be computed by adding all weighted emission probabilities except for $P(O_t|s_j)$ itself:

$$P(O_t|\overline{s_j}) = \sum_{k \neq j} P(O_t|s_k) \cdot P(s_k)$$

So both kinds of definitions lead to a confidence measure, where a low value (i. e. in the case of $C_2$ a negative value) indicates a high degree of certainty for the keyword being correct, whereas a high value point to possible misrecognitions. As an advantage of these calculable confidence scores, neither an additional HMM training (i. e. target & alternate models) nor an artful tuning of other related parameters is necessary. The confidence scores can be computed only using one general HMM set.

The strategy of defining confidence measures, like it is shown above, can be easily incorporated with a conventional HMM-based Viterbi search. Each individual state $s_j$ of all HMMs no longer emits the likelihood $P(O_t|s_j)$, but the local confidence score $c_1$ and $c_2$ respectively. That way, the resulting search procedure can be considered as referring to the changed basic search strategy $W = \mathrm{argmax}_n[C_1(W|O)]$ and $W = \mathrm{argmax}_n\,[C_2(W|O)]$ respectively.

## 2.3. Decoding algorithm

In conventional keyword spotting systems a standard Viterbi search is commonly used to find keyword hypotheses. Two main types can be distinguished: either applying a large vocabulary system or limiting the vocabulary size by using general garbage or filler models.

In order to solve the restrictions concerning out-of-vocabulary speech, the subject of this paper is to introduce novel methods to detect keywords in continuous speech without modeling the non-keyword parts of the speech signal beyond the keyword boundaries by maximizing a confidence measure according to the proposed search strategy. Therefore two different possibilities are discussed and compared.

The **first approach** is similar to the Standard Viterbi algorithm because this method searches for the best path with regard to a optimal summing (integral) confidence score *ISc*:

$$ISc(O) = \sum_{t = t_1}^{t_2} c(O_t|s_j)$$

As a very important point, the variable $t$ ranges from $t_1$ to $t_2$, which are supposed to be the keyword boundaries. Instead of the transition from a preceeding filler model, at every time instant $t$ a new path may start without taking over any previously accumulated scores (i. e. $ISc = 0$). Thus only local confidence scores from the keywords HMM are added and no filler scores are needed. By way of illustration, this is comparable with a filler model always emitting a local confidence score of zero. With $A_j$ being the appropriate transition penalty, the corresponding recursive form of the algorithm can be noted as follows:

$$ISc_{t,\,s_j} = min_j(ISc_{(t-1),\,s_j} + c(O_t|s_j) + A_j)$$

The integral score *ISc* of the last state of the keyword is observed at each frame and serves as an output of the HMM. This score is not a monotonously increasing function. Indicating poor and good matching of the keyword model, it can both increase and decrease because of two reasons. First, the length of the appropriate path (i. e. the number of states) can vary and get lower due to a later timed beginning. Second, the local confidence scores, which serve as individual addends, can be positive as well as negative when using the $C_2$ confidence measure.

The **second approach** is a modified Viterbi algorithm working with length-normalized scores. Because of its foundation on basic principles, that are already described in [1], this algorithm can be regarded as an improved variation of that one described in the above mentioned paper. This method tries to find the best path with regard to a optimal length-normalized (i. e. averaging) confidence score *NSc*:

$$NSc(O) = \frac{1}{T} \cdot \sum_{t = t_1}^{t_2} c(O_t|s_j)$$

Once again, the variable $t$ covers only the time period from $t_1$ to $t_2$, which is elapsed while the keyword is uttered. So the keyword HMM is not concatenated with any predecessor and no filler models are necessary. The second algorithm is more complex, because two variables (i. e. the normalized score $NSc_{t,\,s_i}$ as well as the length of the path $L_{t,\,s_i}$) must be handled for each state $s_j$ and each time instant $t$. The second algorithm can be expressed recursively, too:

$$NSc_{t,\,s_i} = min_j\left(NSc_{(t-1),\,s_j} \cdot \left(1 - \frac{1}{1 + L_{(t-1),\,s_j}}\right) + \frac{1}{1 + L_{(t-1),\,s_j}} \cdot (A_j + c(O_t|s_j))\right)$$

and $$L_{t,\,s_i} = 1 + L_{(t-1),\,s_k}$$

This method computes all possible paths to a certain state $s_j$ in a length-normalized manner and selects the best one. Then the length of the best preceeding state $s_k$ is incremented and stored as the current length. New search paths must be allowed to start at any time instant $t$ with $L_{t,\,s_1} = 1$ . (For more details see [1].) Similar to the first algorithm, in this case the normalized score *NSc* of the last state of the keyword model has to be watched and serves as output value. Again, it can both increase and decrease, indicating poor and good matching of the keyword model.

As a result both methods yield keyword hypotheses for every time instant $t$, that are optimized with regard to the underlaying confidence score. In order to obtain the final keyword hypotheses, the local minima in the course of those frame-level scores have to be extracted and compared to a certain decision threshold. As a post-processing step, possibly competing hypotheses from different keyword models, which have temporal overlappings, can be eliminated by selecting the better one and rejecting the other.

## 3. RESULTS

The proposed methods were examined with the help of the German SpeechDat(M)* database, that was recorded via the public telephone network. The goal was to detect a total number of 47 keywords within so-called application phrases, that are a specific part of the SpeechDat(M)* database. The database was divided into a training set and a test set, where 22136 utterances from 667 speakers were taken for training a general context dependent HMM-set, and a subset of 428 application phrases from different 167 speakers is used for system evaluation.

After the utilization of a preemphasis filter, the 8 kHz sampled speech signal is arranged in overlapping Hamming windowed signal portions of 25 ms length with a frame period of 10 ms. Afterwards, a total number of 24 mel-filtered cepstral coefficients are calculated. In order to compensate different channel transfer characteristics, a maximum likelihood based cepstral mean removal technique is applied to this 24 dimensional vector. Adding 12 first and 12 second order derivatives and including a energy component with its both derivatives, a 51 dimensional vector is composed. By combining two subsequent vectors at each time frame, a 102 dimensional super-vector is obtained, which is transformed using Linear Discriminant Analysis. Finally, the resulting feature vector is determined by selecting the first 24 components out of the transformed and ordered super-vector.

All keyword models are realized by concatenating the corresponding subword units. For this purpose we use context dependent phoneme modeling based on diphones. Each phoneme consists of three segments, whereby the first and the third are dependent on the preceeding and the succeeding phoneme respectively, and the second segment is context independent.

In order to compare the two proposed algorithms in combination with the two different definitions of confidence measures, all four possible variations were evaluated. In table 1 for each case the Figure-of-Merit and the corresponding detection rate at a false alarm rate of 10 fa/h/kw (false alarms per hour per keyword) is shown:

| | standard Viterbi algorithm (*ISc*) | modified Viterbi algorithm (*NSc*) |
|---|---|---|
| confidence score $C_1$ | 38.6% (41.1%) | 71.2% (83.7%) |
| confidence score $C_2$ | 75.0% (83.7%) | 76.4% (86.1%) |

**Table 1:** Figure-of-Merit and (in brackets) the detection rate at a false alarm rate of 10 fa/h/kw.

On the one side, the standard Viterbi algorithm enables a more simple and faster implementation than the modified algorithm, but the modified Viterbi algorithm, optimizing the length-normalized scores, yields some better results. Moreover, the second definition of the confidence score $C_2$, which lead to symmetrical local confidence scores with a balanced value zero, obviously works better than the first approach $C_1$. The former reference system (described in [1]) was also evaluated using the SpeechDat(M) database and yielded a Figure-of-Merit of 63.8% ($fa_{10}$ = 74.7%) on this corpus. So applying the introduced methods provides an up to about 13% improvement of the Figure-of-Merit performance.

## 4. CONCLUSION

A search strategy for detecting keywords in continuous speech is presented which is based on the maximization of a confidence measure. Also the decision on accepting or rejecting a particular keyword hypotheses is made on account of the very same confidence score. Two different possibilities of defining confidence measures are introduced. One is based on the a-posteriori probability of a keyword having been uttered, the other uses a likelihood ratio of a keyword model and an assumed anti-model respectively. As an advantage, no parameter tuning is necessary, because all measures can be calculated. In order to enable a search procedure without the necessity of having HMMs representing the non-keyword speech, two different decoding algorithms are proposed, that enable the optimization of a confidence measure only based on a single keyword model. A comparison of all four possible variations yields as a result, that applying the length-normalized algorithm in conjunction with the second confidence definition $C_2$ works best.

## 5. REFERENCES

1. J. Junkawitsch, L. Neubauer, H. Höge and G. Ruske. A new keyword spotting algorithm with pre-calculated optimal thresholds. Proc. ICSLP, pp. 2067-2070, 1996.
2. M. Weintraub. Keyword-spotting using SRI's DECIPHER large-vocabulary speech-recognition system. Proc. IEEE ICASSP, vol. 2, pp. 463-466, 1993.
3. H. Bourlard, B. D'hoore and J.-M. Boite. Optimizing recognition and rejection performance in wordspotting systems. Proc. IEEE ICASSP, vol. 1, pp. 373-376, 1994.
4. E. Lleida and R. C. Rose. Likelihood ratio decoding and confidence measures for continuous speech recognition. Proc. ICSLP, pp. 478-481, 1996.

(*) see the following URL's:
http://www.phonetik.uni-muenchen.de/SpeechDat.html
http://www.icp.grenet.fr/ELRA/home.html