ITERATIVE POSTERIOR-BASED KEYWORD SPOTTING WITHOUT FILLER MODELS

Marius-Călin Silaghi[†] *and Hervé Bourlard*[†,‡]

[†]Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
[‡]Dalle Molle Institute of Perceptual Artificial Intelligence (IDIAP), Switzerland
{silaghi,bourlard}@lia.di.epfl.ch

ABSTRACT

This paper addresses the problem of detecting keywords in unconstrained speech without explicit modeling of non-keyword segments. The proposed algorithm is based on recent developments in confidence measures using local posterior probabilities, and searches for the segment maximizing the average observation posterior[1] along the most likely path in the hypothesized keyword model.[2] As known, this approach (sometimes referred to as sliding model method) requires a relaxation of the begin/endpoints of the Viterbi matching, as well as a time normalization of the resulting score, making dynamic programming sub-optimal or more complex (more computation and/or more memory).

We present here an alternative (quite simple and efficient) solution to this problem, using an iterative form of Viterbi decoding algorithm, but which does not require scoring for all possible begin/endpoints. Convergence proof of this algorithm is available [8]. Results obtained with this method on 100 keywords chosen at random from the BREF database [5] are reported.

1. INTRODUCTION

This paper addresses the problem of *keyword spotting (KWS)* in unconstrained speech without explicit modeling of non-keyword segments (typically done by using filler HMM models or an ergodic HMM composed of context dependent or independent phone models without lexical constraints). Although several algorithms[3] tackling this type of problem have already been proposed in the past, e.g., by using Dynamic Time Warping (DTW) [4] or Viterbi matching [11] allowing relaxation of the (begin and endpoint) constraints, these are known to require the use of an “appropriate” normalization of the matching scores since segments of different lengths have then to be compared. However, given this normalization and the relaxation of begin/endpoints, straightforward DP is no longer optimal (or, in other words, the DP optimality principle is no longer valid) and has to be adapted, involving more memory and CPU. Indeed, at any possible ending time e, the match score of the best warp and start time b of the reference has to be computed [4] (for all possible start times b associated with unpruned paths). Moreover, in [11], and in the same spirit than what is presented here, for all possible ending times e, the average observation likelihood along the most likely state sequence is used as scoring criterion. Finally, this adapted DP quickly becomes even more complex (or intractable) for more advanced scoring criteria (such as the confidence measures mentioned below).

More recently, work in the field of confidence level, and in the framework of hybrid HMM/ANN systems, it was shown [1] that the use of accumulated local posterior probabilities (as obtained at the output of a multilayer perceptron) normalized by the length of the word segment (or, better, involving a double normalization over the number of phones and the number of acoustic frames in each phone) was yielding good confidence measures and good scores for the re-estimation of N-best hypotheses. Similar work, where this kind of confidence measure was compared to several alternative approaches, was reported in [10] and confirmed this conclusion. However, so far, the evaluation of such confidence measures involved the estimation and rescoring of N-best hypotheses. Similar work and conclusions (also using N-best rescoring) were also reported in using likelihood ratio rescoring and non-keyword rejection [9].

In this paper, we will use a similar scoring technique for keyword spotting without explicit filler model. Compared to previously devised “sliding model” methods (such as [4, 11]), the algorithm proposed here is based on:

1. A matching score defined as the average observation posterior along the most likely state sequence. It is indeed believed that local posteriors (or likelihood ratios, as in [9]) are more appropriate to the task.

2. The iteration of a Viterbi decoding algorithm, which does not require scoring for all begin/endpoints or N-best rescoring, and which can be proved to (quickly) converge to the “optimal”[4] solution without requiring

[1] The accumulated posterior divided by the length of the segment.

[2] It can be easily generalized to more complex matching scores. [8]

[3] Sometimes referred to as “sliding model methods”.

[4] From the point of view of the chosen scoring functions.

any specific filler models, using straightforward Viterbi alignments (similar to regular filler-based KWS, but at the cost of a few iterations).

2. KWS WITHOUT FILLER MODELS

Let $X = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ denote the sequence of acoustic vectors in which we want to detect a keyword, and let M be the HMM model of a keyword M and consisting of L states $\mathcal{Q} = \{q_1, q_2, \ldots, q_\ell, \ldots, q_L\}$. Assuming that M is matched to a subsequence $X_b^e = \{x_b, \ldots, x_e\}$ ($1 \leq b \leq e \leq N$) of X, and that we have an implicit (not modeled) *garbage/filler state* q_G preceding and following M[5], we define (approximate) the log posterior of a model M given a subsequence X_b^e as the average posterior probability along the optimal path, i.e.:

$$\begin{aligned} -\log P(M|X_b^e) &\simeq \frac{1}{e-b+1} \min_{\forall Q \in M} -\log P(Q|X_b^e) \\ &\simeq \frac{1}{e-b+1} \min_{\forall Q \in M} \{-\log P(q^b|q_G) \\ &\quad - \sum_{n=b}^{e-1} [\log P(q^n|x_n) + \log P(q^{n+1}|q^n)] \\ &\quad - \log P(q^e|x_e) - \log P(q_G|q^e)\} \end{aligned} \quad (1)$$

where $Q = \{q^b, q^{b+1}, ..., q^e\}$ represents one of the possible paths of length $(e - b + 1)$ in M, and q^n the HMM state visited at time n along Q, with $q^n \in \mathcal{Q}$. In this expression, q_G represents the "garbage" (filler) state which is simply used here as the non-emitting initial and final state of M. Transition probabilities $P(q^b|q_G)$ and $P(q_G|q^e)$ can be interpreted as the keyword entrance and exit penalties, as optimized in [3], but these have not been optimized here. In our case, local posteriors $P(q_\ell|x_n)$ were estimated as output values of a multilayer perceptron (MLP) used in a hybrid HMM/ANN system [2].

For a specific sub-sequence X_b^e, expression (1) can easily be estimated by dynamic programming since the subsequence and the associated normalizing factor $(e - b + 1)$ are given. However, in the case of keyword spotting, this expression should be estimated for all possible begin/endpoint pairs $\{b, e\}$ (as well as for all possible word models), and we define the matching score of X on M as:

$$S(M|X) = -\log P(M|X_{b^*}^{e^*}) \quad (2)$$

where the optimal begin/endpoints $\{b^*, e^*\}$, and the associated optimal path Q^*, are the ones yielding the lowest average local posterior:

$$\langle Q^*, b^*, e^* \rangle = \operatorname*{argmin}_{\{Q,b,e\}} \frac{-1}{e-b+1} \log P(Q|X_b^e) \quad (3)$$

Of course, in the case of several keywords, all possible models will have to be evaluated.

As shown in [1, 10], a double averaging involving the number of frames per phone and the number of phones will usually yield slightly better performance:

$$\langle Q^*, b^*, e^* \rangle = \quad (4)$$
$$\operatorname*{argmin}_{\{Q,b,e\}} \frac{-1}{J} \sum_{j=1}^{J} \left(\frac{1}{e_j - b_j + 1} \sum_{n=b_j}^{e_j} \log P(q_j^n|x_n) \right)$$

where J represents the number of phones in the hypothesized keyword model and q_j^n the hypothesized phone q_j for input frame x_n.

However, given the time normalization and the relaxation of begin/endpoints, straightforward DP is no longer optimal and has to be adapted, usually involving more memory and CPU. A new (and simple) solution to this problem will be proposed in Section 4.

3. FILLER-BASED KWS

Although various solutions have been proposed towards the direct optimization of (2) as, e.g., in [4, 11], most of the keyword spotting approaches today prefer to preserve the optimality and simplicity of Viterbi DP by modeling the complete input [6] and explicitly [7] or implicitly [3] modeling non-keyword segments by using so called filler or garbage models as additional reference models. In this case, we assume that non-keyword segments are modeled by extraneous garbage models/states q_G (and grammatical constraints ruling the possible keyword/non-keyword sequences).

In this paper, we will consider only the case of detecting one keyword per utterance at a time. In this case, the keyword spotting problem amounts at matching the whole sequence X of length N onto an extended HMM model $\overline{M}$ consisting of the states $\{q_G, q_1, \ldots, q_L, q_G\}$, in which a path (of length N) is denoted $\overline{Q} = \{\overbrace{q_G, ...q_G}^{b-1}, q^b, q^{b+1}, ..., q^e, \overbrace{q_G, ...q_G}^{N-e}\}$ with $(b-1)$ garbage states q_G preceding q^b and $(N - e)$ states q_G following q^e, and respectively emitting the vector sequences X_1^{b-1} and X_{e+1}^N associated with the non-keyword segments.

Given some estimation of $P(q_G|x_n)$ (e.g., using probability density functions trained on non keyword utterances), the optimal path $\overline{Q}^*$ (and, consequently b^* and e^*) is then given by:

$$\begin{aligned} \overline{Q^*} &= \operatorname*{argmin}_{\forall \overline{Q} \in \overline{M}} -\log P(\overline{Q}|X) \\ &= \operatorname*{argmin}_{\forall \overline{Q} \in \overline{M}} \{-\log P(Q|X_b^e) \\ &\quad - \sum_{n=1}^{b-1} \log P(q_G|x_n) - \sum_{n=e+1}^{N} \log P(q_G|x_n)\} \end{aligned} \quad (5)$$

[5] Thus implicitly introducing the grammatical constraint that we have only one keyword, preceded and followed by a non-keyword segment.

which can be solved by straightforward DP (since all paths have the same length). The main problem of filler-based keyword spotting approaches is then to find ways to best estimate $P(q_G|x_n)$ in order to minimize the error introduced by the approximations. In [3], this value was defined as the average of the N best local scores while, in other approaches, this value is generated from explicit filler HMMs. However, these approaches will usually not lead to the "optimal" solution given by (2).

4. ITERATING VITERBI DECODING (IVD)[6]

In the following, we show that it is possible to define an iterative process, referred to as *Iterating Viterbi Decoding (IVD)*[6] with good/fast convergence properties, estimating the value of $P(q_G|x_n)$ such that straightforward DP (5) yields exactly the same segmentation (and recognition results) than (3). While the same result could be achieved through a modified DP in which all possible combinations (all possible begin/endpoints) would be taken into account, it is possible to show that the algorithm proposed below is more efficient (in terms of both CPU and memory requirements).

The IVD algorithm is based on the same criterion than the filler based approaches (5), but rather than looking for explicit (and empirical) estimates of $P(q_G|x_n)$ we aim at mathematically estimating its value (which will be different and adapted to each utterance) such that solving (5) is equivalent to solving (3). Thus, we perform an iterative estimation of $P(q_G|x_n)$, such that the segmentation resulting of (5) is the same than what would be obtained from (3).

Defining $\varepsilon = -\log P(q_G|x_n)$, the proposed algorithm can be summarized as follows:

1. Start from an initial value $\varepsilon_0 = \varepsilon$[7], (e.g., with ε equal with a cheap estimation of the score of a "match"). In the experiments reported below, ε was initialized to $-\log$ of the maximum of the local probabilities $P(q_k|x_n)$ for each frame x_n.

   An alternative choice could be to initialize ε_0 to a predefined score that expression (1) should reach to declare a keyword "matching" (see point 4 below). In this last case, if ε increases at the first iteration, then we can (as proven) directly infer that the match will be rejected, otherwise it will be accepted.

2. Given the current estimate ε_t of $P(q_G|x_n)$ at iteration t, find the optimal path $\langle \overline{Q}_t, b_t, e_t \rangle$ according to (5) and matching the complete input.

3. Update ($t = t + 1$) the estimated value of ε_t, defined as the average of the local posteriors along the optimal path Q_t (matching the $X_{b_t}^{e_t}$ resulting of (5) on the keyword model) i.e.:

$$\varepsilon_{t+1} = -\frac{1}{(e_t - b_t + 1)} \log P(Q_t|X_{b_t}^{e_t}) \qquad (6)$$

4. Return to (2) and iterate until convergence. If we are not interested in the optimal segmentation, this process could also be stopped as soon as ε reaches a (predefined) minimum threshold below which we can declare that a keyword has been detected.

Convergence proof of this process and generalization to other criteria, are given in [8]: each IVD iteration (from the second iteration) will decrease the value of ε_t, and the final path yields the same solution than (3).

5. EXPERIMENTAL RESULTS

Preliminary tests of the IVD algorithm were performed on the BREF database [5], a continuous, read speech microphone database. As done in [1], $3,736$ utterances were used for training an artificial neural network (multilayer perceptron) to generate local (context-independent) phone posterior probabilities. 242 utterances (with a $2,300$ word lexicon), from which *100 keywords* were selected at random, were used for testing. These keywords were simply represented by simple hybrid HMM/ANN models [2] based on context-independent phones.

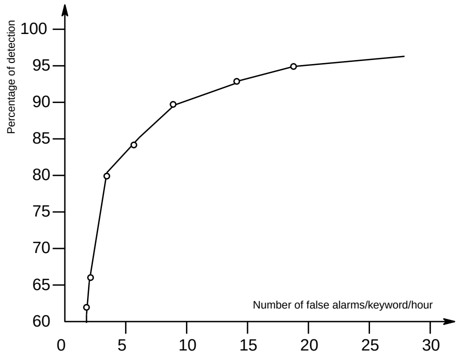

Figure 1: *ROC of the IVD-based keyword detection based on (2) as a function of number of false alarms/keyword/hour, as obtained on 242 BREF test sentences and 100 keywords selected at random.*

The resulting ROC (Receiver Operating Characteristics) curve, using IVD to estimate (2), is presented in Figure 1 and shows good performance compared to similar experiments [3], although no parameters (such as keyword entrance penalties) were tuned to optimize performance. For

[6] Patent pending.

[7] In [8], it is actually proven that the iterative process presented here will always converge to the same solution (in more or less cycles, with the worst case upper bound of N iterations) independently of this initialization.

computing the segmentation, 3 to 5 iterations were needed. If the segmentation is not needed, the “matching” decision can be taken with only one iteration as described in the initialization step of the algorithm.

For comparison, the ROC curve obtained (for the same keywords and test sentences) with criterion (4), involving a double normalization, is reported in Figure 2. As also reported for confidence measure rescoring [1], this measures is yielding even better KWS performance.

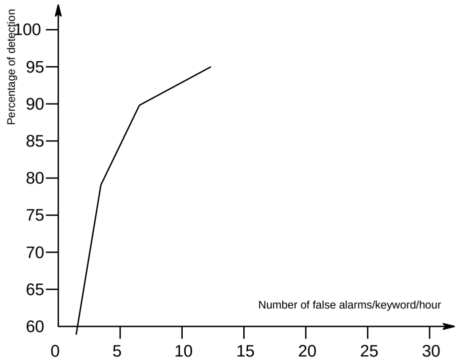

Figure 2: *ROC using criterion (4) (double normalization), on 242 BREF test sentences containing 100 keywords selected at random.*

6. CONCLUSIONS

In this paper, we have thus proposed a new method for keyword spotting, based on recent advances in confidence measures, using local posterior probabilities, but without requiring the explicit use of filler models.

A new algorithm, referred to as *Iterating Viterbi Decoding (IVD)*, to solve the above optimization problem with a simple DP process (not requiring to store pointers and scores for all possible ending and start times), at the cost of a few iterations.

While the proposed approach allows for an easy generalization to more complex criteria, preliminary results obtained on the basis of 100 keywords (and without any specific tuning) appear to be particularly competitive to other alternative approaches.

7. ACKNOWLEDGMENTS

We thank Giulia Bernardis for helping us with the BREF database and providing us with a neural network trained for this task. We also acknowledge the useful discussions with Prof. Boi Faltings. The authors were partly supported by the Swiss Federal Office of Education and Science (OFES) through the THISL European LTR project.

8. REFERENCES

[1] Bernardis, G. and Bourlard, H., “Improving posterior-based confidence measures in hybrid HMM/ANN speech recognition systems,” *Proceedings of Intl. Conf. on Spoken Language Processing* (Sydney, Australia), pp. 775-778, 1998.

[2] Bourlard, H. and Morgan, N., *Connectionist Speech Recognition - A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[3] Bourlard, H., D’hoore, B., and Boite, J.-M., “Optimizing recognition and rejection performance in wordspotting systems,” *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing* (Adelaide, Australia), pp. I:373-376, 1994.

[4] Bridle, J.S., “An efficient elastic-template method for detecting given words in running speech,” *Proc. of the Brit. Acoust. Soc. Meeting*, pp. 1-4, April 1973.

[5] Lamel, L.F., Gauvain, J-L., and Eskénazi, M., “BREF, a large vocabulary spoken corpus for French,” *Proceedings of Eurospeech’91*, pp. 505-508, 1991.

[6] Rohlicek, J.R., “Word spotting,” in *Modern Methods of Speech Processing*, R.P. Ramachandran and R. Mammone (Eds.), Kluwer Academics Publishers, pp. 123-157, 1995.

[7] Rose, R.C. and Paul, D.B., “A hidden Markov model based keyword recognition system,” *Proc. of ICASSP’90*, pp. 129-132, 1990.

[8] Silaghi, M.-C. and Bourlard H., “Posterior-Based Keyword Spotting Approaches Without Filler Models,” *Swiss Federal Institute of Technology Lausanne (EPFL)*, Technical Report, 1999.

[9] Sukkar, R.A. and Lee, C.-H., “Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition,” *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 6, pp. 420-429, 1996.

[10] Williams, G. and Renals, S., “Confidence measures for hybrid HMM/ANN speech recognition,” *Proceedings of Eurospeech’97*, pp. 1955-1958, 1997.

[11] Wilpon, J.G., Rabiner, L.R., Lee C.-H., and Goldman, E.R., “Application of hidden Markov models of keywords in unconstrained speech,” *Proc. of ICASSP’89*, pp. 254-257, 1989.