# KEYWORD SPOTTING USING DURATIONAL ENTROPY

*Jitendra Ajmera and Florian Metze*

Deutsche Telekom Laboratories
Berlin, Germany

## ABSTRACT

This paper deals with the task of detection of a given keyword in continuous speech. We build upon a previously proposed algorithm where a modified Viterbi search algorithm is used to detect keywords, without requiring any explicit garbage or filler models. In this work, the concept of durational entropy is used to further discard a large fraction of false alarm errors. Durational entropy is defined as the entropy of the distribution of state occupancies. A method to recursively compute it for all Viterbi paths is also presented in this paper. Experimental results on one hour of broadcast news data suggest that durational entropy constraints can indeed be used to avoid a large number of false alarms errors at a minimal cost of degradation in keyword detection accuracy.

***Index Terms***— Hidden Markov models (HMM), Viterbi decoding, entropy, maximum likelihood decoding, speech recognition.

## 1. INTRODUCTION

Keyword spotting has remained an interesting and challenging research task for some time. Although quite similar to speech recognition in nature, the major challenge in keyword spotting is rejection of non-keyword (out-of-vocabulary (OOV) items in continuous speech recognition (CSR) terminology) events. The modeling of these non-keyword events is generally referred to as garbage or filler modeling in literature.

Many hidden Markov model (HMM) based approaches have been proposed in literature where an explicit garbage model is used [1, 2, 3]. There also have been other approaches which do not employ explicit garbage model but rely on a non-parametric representation of non-keyword events [3, 4].

In HMM based approaches with a garbage model, Viterbi algorithm [5] is generally used to find segmentation in terms of garbage events and keyword. To find this segmentation, one requires information about begin and end-points of the utterance, which is generally assumed available in most of the previous work. Also, some of these techniques have impractical heuristical assumptions such as at most one occurrence of a keyword per utterance [4].

This work builds upon previous work in [6] which takes care of some of the shortcomings mentioned above. Some advantages of this algorithm are:

- No filler or garbage model required, so there are no parameters or hyper-parameters (in case of non-parametric representation) to be learnt from training data.
- No backtracking required, so no knowledge of begin and end points of the utterance are required and any keyword can be located as and when it occurs in a time-synchronous manner.
- The algorithm needs very low computational and memory resources.

In this work, we propose using durational entropy as an additional cue toward reliability of a hypothesized instance of a keyword. Here, the durational entropy is defined as entropy of the distribution of state occupancies (number of frames associated with each HMM state in a Viterbi path). The basic idea is not only to look at likelihood score of a Viterbi path but also analyze how well each state has contributed in building up that score. Resulting durational entropy will be lower for more uniform contributions of different states to a log-likelihood score. In our experiments (Figure 2), this entropy is generally significantly lower for correctly detected instances as compared to that of false alarm situations.

This paper is organized as follows: Section 2 presents a brief review of the basic keyword algorithm proposed in [6]. Section 3 then defines durational entropy and presents a method to compute this entropy, recursively, for every Viterbi path in the forward course of the Viterbi algorithm. Section 4 presents the experimental setup and also discusses thresholding issues.

## 2. KEYWORD SPOTTING ALGORITHM

The algorithm is based on hidden Markov model (HMM) representation of a keyword. It is generally achieved by concatenating HMMs of underlying phonemes or other subword units in a left-to-right fashion. The Viterbi algorithm has to be modified for this algorithm since it does not involve any garbage or filler model (or competing hypothesis). This is done as follows:

The algorithm is based on local scores which are interpreted as distance measure from a certain state to the best state. Local score $LSc_t^{s_j}$ of state $s_j$ at time $t$ is computed as:

$$LSc_t^{s_j} = \log p(x_t|s_j) - \max_{i=1}^{J} \log p(x_t|s_i) \quad (1)$$

where, $J$ is the number of states in the keyword-HMM.

If $a_{i,j}$ is transition probability of going from state $s_i$ to state $s_j$, the recursive search equations can be written as:

$$NSc_t^{s_j} = \max_i \frac{NSc_{t-1}^{s_i} L_{t-1}^{s_i} + \log a_{i,j} + LSc_t^{s_j}}{1 + L_{t-1}^{s_i}} \quad (2)$$

where, $L_t^{s_i}$ and $NSc_t^{s_i}$ denote the length of a path leading to state $s_i$ at time $t$ and resulting normalized score, respectively. If the best predecessor state to state $s_j$ in above equation is denoted as $s_{best}$, then $L_t^{s_j} = L_{t-1}^{s_{best}} + 1$.

At any time, a new path may start with $L_t^{s_1} = 1$ and $NSc_t^{s_1} = LSc_t^{s_1}$.

The normalized score at the last state of the HMM $NSc_t^{s_J}$ ($J$ being number of states) is then compared against a threshold to make decision about existence of the keyword. A way to compute word-specific threshold was also proposed in [6].

## 3. DURATIONAL ENTROPY

Let us consider a Viterbi path spanning observation sequence of length $L$, such that each state ($s_j, j = 1...N$) of the HMM occupies $L_j$ observations and $L = \sum_{j=1}^{J} L_j$. The durational entropy of this path can then be calculated as:

$$dE = \frac{\sum_{j=1}^{J} \frac{L_j}{L} \log \frac{L_j}{L}}{\log J} \quad (3)$$

Durational entropy for a word computed as (3) is bounded between 0 and -1.

The segmentation of observation sequence (which observation corresponds to which state) is generally derived in the backtracking part of the Viterbi algorithm. If we want to incorporate durational entropy in the keyword spotting algorithm, this would mean that we will have to derive backtracking at every time $t$, and therefore will also require backtracking pointers which are missing in above-mentioned algorithm.

In the following, we propose a mechanism by which durational entropy can be computed for every Viterbi path in a recursive manner exactly like the computation of accumulated normalized scores $NSc_t^{s_j}$.

Let us denote the durational entropy of the path leading to state $s_j$ at time $t$ as $E_t^{s_j}$. Also, let $N_j$ denote the number of times state $s_j$ has been visited in this path. In a strict left-to-right topology without skips, this can be computed recursively as:

If $s_{best}$, as given by Eq. 2 is same as $s_j$: $N_{s_j} = N_{s_j} + 1$, and

$$E_t^{s_j} = E_{t-1}^{s_j} - (N_{s_j} - 1.0) log(N_{s_j} - 1.0) + N_{s_j} log N_{s_j} \quad (4)$$

Otherwise: $N_{s_j} = 1$, and $E_t^{s_j} = E_{t-1}^{s_{best}}$.

The durational entropy of the word at time $t$ is then computed as:

$$dE_t = \frac{\frac{E_t^{s_J}}{L_t^{s_J}} - \log L_t^{s_J}}{\log J} \quad (5)$$

Durational entropy provides a simple measure of how different states are occupied across the length ($L_t^{s_J}$) of a word. In the proposed approach, normalized score $NSc_t^{s_J}$ as well as durational entropy $dE_t$ (as given by Eq.5) are tracked and compared against respective thresholds. What we claim by doing so is that although it is important to consider log likelihoods (as done in most speech recognition and related technologies), it is also important at the same time to analyze how well each state of the word has contributed to this likelihood score. A similar study based on durational modeling was presented in [7]. Essentially, we would like to avoid a situation in which a set of specific states find a very good match to acoustic data and produce very high likelihood scores.

Figure 1 presents a picture of how normalized score $NSc_t^{s_J}$ (Eq. 2), length of the word $L_t^{s_J}$ (Eq. **??**) and durational entropy $dE_t$ (Eq. 5) evolve over time. There are two correct instances of the keyword "country" and two false alarms marked by 'C' and 'F' in this figure. The two false alarm situations shown in the figure correspond to words *currency* and *story*. A sharp decline in the length-pane of the figure marks time instants when a new path is revealed at the last state, as discussed in Section 2. The horizontal dashed lines in first and last panes show potential thresholds that can be used to make keyword decisions.

The figure shows that it is difficult to avoid the two false alarms by using only log-likelihood statistics even by using best threshold. These two false alarm situations can be avoided by considering durational entropy in combination with log-likelihood statistics. The dashed line in the durational entropy pane shows a potential threshold value which would reject these two false alarms. Moreover, this threshold does not have to be tuned for each individual keyword. All the experiments reported in this paper are done using same value of entropy threshold across keywords.

## 4. EXPERIMENTS AND EVALUATION

The keyword spotting algorithm mentioned in Sections 2 and 3 was tested on one hour of American broadcast news data. The speech data in these sets is sampled at 16KHz. There are over 50 different speakers, both male and female, present in the data-sets.

Mel Frequency Cepstral Coefficients (MFCC) [8] were extracted from the speech signal. This together with first and

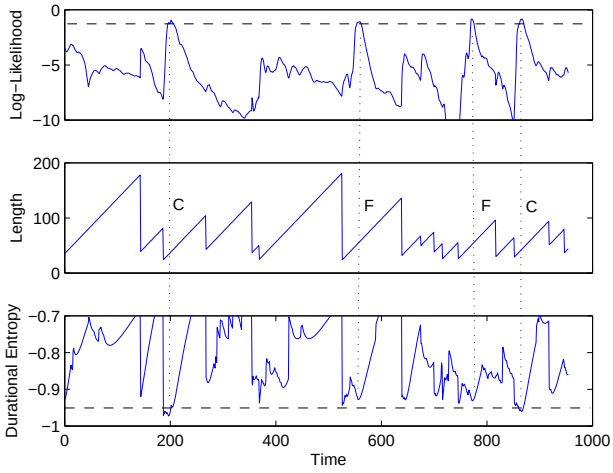


**Fig. 1**. Evolution of normalized log-likelihood score ($NSc_t^{s_J}$), length of the keyword "country" ($L_t^{s_J}$) and durational entropy $dE_t$ over time $t$. There are two correct detections (marked by 'C' in the middle pane) and two false alarms (marked by 'F'). The horizontal dashed lines in first and third panes show potential score and entropy thresholds that can be used to make keyword detection decisions. A sharp decline in length-pane shows a new Viterbi path.

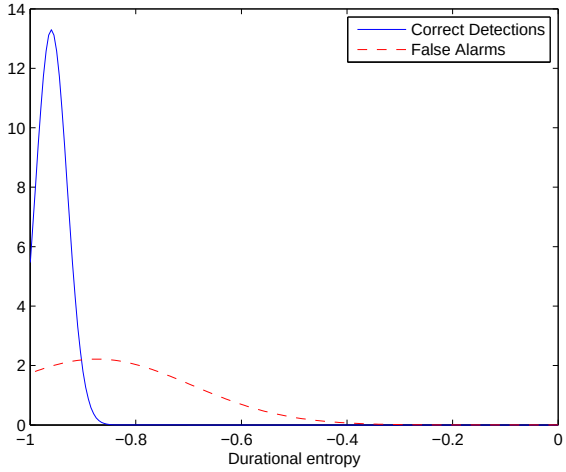


**Fig. 2**. Distributions of durational entropies for correct and false alarm errors for the keyword spotting task.

second order delta coefficients resulted in a 42 dimensional feature vector every 10ms.

Acoustic models used for these experiments were trained using Janus toolkit [9]. These acoustic models correspond to context dependent phonemes subword units.

A total of 15 keywords were picked based on their number of occurrences and significance. These keywords are *medical*, *airlines*, *investigation*, *government*, *country*, *president*, *american*, *palestinians*, *consumer*, *business*, *united*, *white house*, *campaign*, *information* and *mediation*. A total of 138 occurrences of these keywords are present in the data.

### 4.1. Thresholding

As mentioned above, normalized score $NSc_t^{s_J}$ and durational entropy $dE_t$ (Eq. 5) are tracked and compared against their respective thresholds at every time instant. In the following, we discuss how these thresholds can be derived.

In an ideal situation, in which every state turns out to be best (Eq.1) in its respective position in the Viterbi path, the normalized score at the last state of the word $NSc_t^{s_J}$ would simply be a summation of logarithms of transition probabilities and can be deterministically computed.

In practice, log-likelihoods of other subword unit states would be competing against and many times may turn out to be better than the *ideal state* [1]. These situations will penalize the likelihood score. A threshold, thus, should be based on degree of this penalty for each keyword.

Probability of competition and confusion across states of a word (which finally results in penalization) should grow with the number of states ($J$) comprising keyword-HMM, and length of the word. Since the likelihood score is already normalized by length of the word (Eq. 2), the likelihood threshold was simply chosen to be proportional to the number of states in this work. Thus, $T_{score} = T_{trans} + K \cdot J$, where $T_{trans}$ is the deterministic term based on transition probabilities.

Although the derivation of the threshold for duration entropy $T_{entropy}$ is heuristic, the fact that it is naturally normalized between 0 and -1 and that it does not depend on number of states or length of the word, makes it relatively easy and intuitive. An optimum threshold, however, depends on the choice of subword units (monophones, syllables, etc.) and associated acoustic model topology. An analysis of durational entropies for correct detections (Figure 2) can provide a threshold for durational entropy where a lot of false alarms can be avoided at the cost of minimal degradation in keyword detection accuracy.

Figure 2 presents distributions of durational entropies of correctly detected and false alarm errors corresponding to the current task of keyword spotting. These distributions correspond to nearly 110 correctly detected instances and 140 false alarm errors. Again, this figure shows that durational entropy for correctly detected instances is much lower compared to those of false alarm errors.

[1] *ideal state* refers to the state which will be part of the best Viterbi path through the length of the word

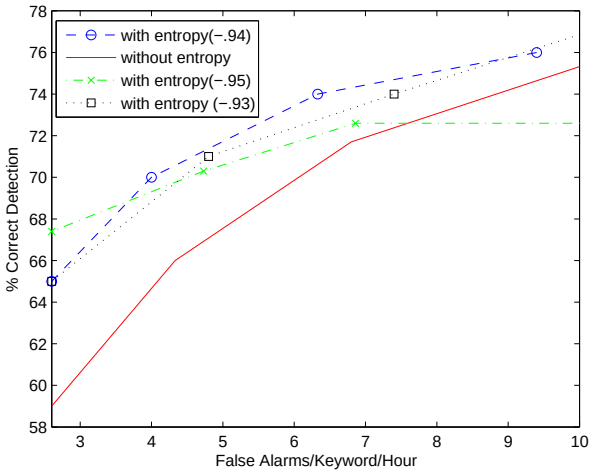


**Fig. 3**. DET curves for baseline (dashed, without entropy) and proposed (solid, with entropy) configurations. Both the curves are obtained by changing the value of threshold for normalized score $T_{score}$, while using a constant value of $T_{entropy}$ for proposed configuration.

### 4.2. Evaluation

Figure 3 shows the decision error tradeoff (DET) curves for both baseline (solid curve) and proposed configurations (dashed curves). In the baseline configuration, a keyword is detected when $NSc_t^{sJ} > T_{score}$. In the proposed configuration, a keyword is detected when $NSc_t^{sJ} > T_{score}$ as well as $dE_t < T_{entropy}$. Each curve in this figure is obtained by varying the value of $K$, while keeping the value of $T_{entropy}$ constant across keywords and across different values of $K$. Performance with 3 different values of $T_{entropy}$ is shown in this figure 3.

The figure shows the efficiency of the proposed approach of using durational entropy constraints in combination with log-likelihood statistics. Durational entropy constraints indeed are very effective at avoiding false alarms. For example, at $K = .025$, using $T_{entropy} = -0.94$ avoided 64 false alarm instances at the cost of missing 3 correct detections. At the same value of $K$, $T_{entropy} = -0.95(-0.93)$ avoided 88 (*48*) false alarms, at the cost of missing 8 (*3*) correct detections.

It should be noted that the basic keyword spotting algorithm (modified Viterbi algorithm, Section 2) is still driven by log-likelihood scores. Thus, an upper limit on the keyword detection accuracy would still be decided by effectiveness of the underlying acoustic models. The durational entropy only provides a framework to analyze relative behavior of different states in building a keyword hypothesis and thus helps avoid false alarm errors.

## 5. CONCLUSIONS

This paper used the concept of durational entropy in a keyword spotting framework to avoid false alarm errors. Durational entropy was defined as entropy of the distribution of state occupancies in a Viterbi path. A method to recursively compute durational entropy in time synchronous Viterbi decoding was also presented in this paper. It was established that durational entropies for correct instances are generally lower than those of false alarms. Experimental results on one hour of broadcast news data showed that durational entropy can help avoid a large number of false alarms at a cost of minimal degradation in detection accuracy.

## 6. REFERENCES

[1] J. G. Wilpon, L. R. Rabiner., C. H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hmm's," *IEEE Trans. On ASSP*, vol. 38, no. 11, pp. 1870–1878, 1990.

[2] R. C. Rose and D. B. Paul, "A hidden markov model based keyword recognition system," in *ICASSP*, 1990, pp. 129–132.

[3] J. M. Boite, H. Boulard, B. D'hoore, and M. Haesen, "A new approach towards keyword spotting," in *Eurospeech*, 1993, vol. 2, pp. 1273–1276.

[4] M. Silaghi and H. Bourlard, "Iterative posterior-based keyword spotting without filler models," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU'99) Workshop*, 1999.

[5] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, February 1989, vol. 77, pp. 257–286.

[6] J. Junkawitsch, L. Neubauer, H. Höge, and G. Ruske, "A new keyword spotting algorithm with pre-calculated optimal thresholds," in *ICSLP*, Philadelphia, PA, 1996, vol. 4, pp. 2067–2070.

[7] B. H. Juang, L. R. Rabiner, and S. E. Levinson, "Recent developments in the application of hidden markov models to speaker independent isolated word recognition," in *ICASSP*, 1985, pp. 9–12.

[8] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyl- labic word recognition in continuously spoken sentences," *IEEE Trans. on ASSP*, pp. 357–366, 1980.

[9] M.Finke, P.Geutner, H.Hild, T.Kemp, K.Ries, and M.Westphal, "The karlsruhe verbmobil speech recognition engine," in *ICASSP*, 1997.