Discriminative Keyword Spotting

Joseph Keshet [a,*], David Grangier [b], Samy Bengio [c]

[a] *IDIAP Research Institute, Rue Marconi 19, CH-1920 Martigny, Switzerland*

[b] *NEC Labs America, 4 Independence Way, Princeton, NJ, 08540*

[c] *Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA, 94043*

Abstract

This paper proposes a new approach for keyword spotting, which is based on large margin and kernel methods rather than on HMMs. Unlike previous approaches, the proposed method employs a discriminative learning procedure, in which the learning phase aims at achieving a high area under the ROC curve, as this quantity is the most common measure to evaluate keyword spotters. The keyword spotter we devise is based on mapping the input acoustic representation of the speech utterance along with the target keyword into a vector space. Building on techniques used for large margin and kernel methods for predicting whole sequences, our keyword spotter distills to a classifier in this vector-space, which separates speech utterances in which the keyword is uttered from speech utterances in which the keyword is not uttered. We describe a simple iterative algorithm for training the keyword spotter and discuss its formal properties, showing theoretically that it attains high area under the ROC curve. Experiments on read speech with the TIMIT corpus show that the resulted discriminative system outperforms the conventional context-independent HMM-based system. Further experiments using the TIMIT trained model, but tested on both read (HTIMIT, WSJ) and spontaneous speech (OGI-Stories), show that without further training or adaptation to the new corpus our discriminative system outperforms the conventional context-independent HMM-based system.

Key words: keyword spotting, spoken term detection, speech recognition, large margin and kernel methods, support vector machines, discriminative models

* Corresponding author.
Email addresses: `jkeshet@idiap.ch` (Joseph Keshet), `dgrangier@nec-labs.com` (David Grangier), `bengio@google.com` (Samy Bengio).

1 Introduction

Keyword spotting refers to the detection of all occurrences of any given word in a speech signal. Most previous work on keyword spotting has been based on hidden Markov models (HMMs). See for example (Benayed et al., 2004; Ketabdar et al., 2006; Silaghi and Bourlard, 1999; Szoke et al., 2005) and the references therein. Despite their popularity, HMM-based approaches have several known drawbacks such as convergence of the training algorithm (EM) to a local maxima, conditional independence of observations given the state sequence and the fact that the likelihood is dominated by the observation probabilities, often leaving the transition probabilities unused. However, the most acute weakness of HMMs for keyword spotting is that they do not aim at maximizing the detection rate of the keywords directly.

In this paper we propose an alternative approach for keyword spotting that builds upon recent work on discriminative large margin and kernel methods, trying to overcome some of the inherent problems of the HMM approaches. Our approach solves directly the keyword spotting problem rather than using a large vocabulary speech recognizer (as in Szoke et al., 2005), and does not estimate a garbage or background model (as in Silaghi and Bourlard, 1999). The advantage of margin-based discriminative approaches stems from the fact that the objective function used during the learning phase is tightly coupled with the decision task one needs to perform. In addition, there is both theoretical and empirical evidence that large margin strategies are likely to outperform generative models for the same task (see for instance Cristianini and Shawe-Taylor, 2000; Vapnik, 1998). One of the main goals of this work is to extend the notion of discriminative large margin and kernel methods to the task of keyword spotting.

Our proposed method is based on recent advances in kernel machines and large margin classifiers for sequences (Shalev-Shwartz et al., 2004; Taskar et al., 2003), which in turn build on the pioneering work of Vapnik and colleagues (Cristianini and Shawe-Taylor, 2000; Vapnik, 1998). The keyword spotter we devise is based on mapping the speech signal along with the target keyword into a vector-space endowed with an inner-product. Our learning procedure distills to a classifier in this vector-space which is aimed at separating the utterances that contain the keyword from those that do not contain it. On this aspect, our approach is hence related to support vector machine (SVM), which has already been successfully applied in speech applications (Keshet et al., 2001; Salomon et al., 2002). However, the model proposed in this paper differs significantly from a classical SVM due to the sequential nature of the keyword spotting problem.

This paper is organized as follows. In Section 2, we formally introduce the keyword spotting problem. We then present the large margin approach for keyword spotting in Section 3. Next, the proposed iterative learning method is described in Section 4. In Section 5, we describe the efficient evaluation of our keyword spotter and its complexity. Our method is based on non-linear phoneme recognition and segmentation functions. The specific feature functions we use for are presented in Section 6. In Section 7, we present experimental results. We conclude the paper in Section 8.

Related Work. Most work on keyword spotting has been based on HMMs. In these approaches, the detection of the keyword is based on an HMM composed of two sub-models, the *keyword model* and the background or *garbage model*, such as the HMM depicted in Figure 6. Given a speech sequence, such a model detects the keyword through Viterbi decoding: the keyword is considered as uttered in the sequence if the best path goes through the keyword model. This generic framework encompasses the three main classes of HMM-based keyword spotters, that is *whole-word* modeling, *phonetic-based* approaches and *large-vocabulary-based* approaches.

Whole-word modeling is one of the earliest approaches using HMM for keyword spotting (Rahim et al., 1997; Rohlicek et al., 1989). In this context, the keyword model is itself an HMM, trained from recorded utterances of the keyword. The garbage model is also an HMM, trained from non-keyword speech data. The training of such a model hence requires several recorded occurrences of the keyword, in order to estimate reliably the keyword model parameters. Unfortunately, in most applications, such data are rarely provided for training, which yields the introduction of phonetic-based word spotters.

In phonetic-based approaches, both the keyword model and the garbage model are built from phonemes (or triphones) sub-models (Bourlard et al., 1994; Manos and Zue, 1997; Rohlicek et al., 1993). Basically, the keyword model is a left-right HMM, resulting from the concatenation of the sub-models corresponding to the keyword phoneme sequence. The garbage model is an ergodic HMM, which fully connects all phonetic sub-models. In this case, sub-model training is performed through embedded training from a large set of acoustic sequences labeled phonetically, like for speech recognition HMMs (Rabiner and Juang, 1993). This approach hence does not require training utterances of the keyword, solving the main limitation of the whole word modeling approach. However, the phonetic-based HMM has another drawback, due to the use of the same sub-models in the keyword model and in the garbage model. In fact, the garbage model can intrinsically model any phoneme sequence, including the keyword itself. This issue is typically addressed by tuning the prior probability of the keyword, or by using a more refined garbage model, e.g. Bourlard et al. (1994); Manos and Zue (1997). Another solution can also be to avoid the need for garbage modeling through the computation of the likelihood of the keyword model for any subsequence of the test signal, as proposed in Junkawitsch et al. (1997).

A further extension of HMM spotter approaches consists of using Large Vocabulary Continuous Speech Recognition (LVCSR) HMMs. This approach can actually be seen as a phonetic-based approach in which the garbage model only allows valid words from the lexicon, except the targeted keyword. This use of additional linguistic constraints is shown to improve the spotting performance (Cardillo et al., 2002; Rose and Paul, 1990; Szoke et al., 2005; Weintraub, 1995). Such an approach however raises practical concerns: one can wonder whether the design of a keyword spotter should require the expensive collection a large amount of labeled data typically needed to train LVCSR systems, as well as the computational cost implied by large vocabulary decoding (Manos and Zue, 1997).

Over the last years, significant effort toward discriminative training of HMMs has been proposed as an alternative to likelihood maximization (Bahl et al., 1986; Juang et al., 1997; Fu and Juang, 2007). These training approaches aim at both maximizing the probability of the correct transcription given an acoustic sequence, and minimizing the probability of the incorrect transcriptions given an acoustic sequence. When applied to keyword spotting, none of these approaches closely tie the training objective with a final spotting objective, such as maximizing the area under the Receiver Operating Curve. In our approach, we reach this goal by proposing a discriminative model focusing on an adequate criterion. In this sense, our work significantly differs from discriminative HMM training for speech recognition, as our learning procedure directly focuses on the spotting performance. Furthermore, we do not constrain the underlying model to be probabilistic, which allows a greater freedom in selecting the set of features.

2 Problem Setting

In the keyword spotting task, we are provided with a speech utterance and a keyword and the goal is to identify whether the keyword is uttered in the speech utterance and where. Any keyword (or word) is naturally composed of a sequence of phonemes. Hence we can state the goal as to detect whether the corresponding phoneme sequence is articulated in the given utterance and where. We assume that the goal of the keyword spotting refers to the detection of *any* keyword, and not only to the keywords already seen in the training phase. In what follows, we also assume that the utterance is small enough for the keyword to be articulated only *once*. If the utterance is longer than that, we apply the keyword spotter on a sliding window of an appropriate length.

In this section we formally describe the keyword spotting problem. Throughout

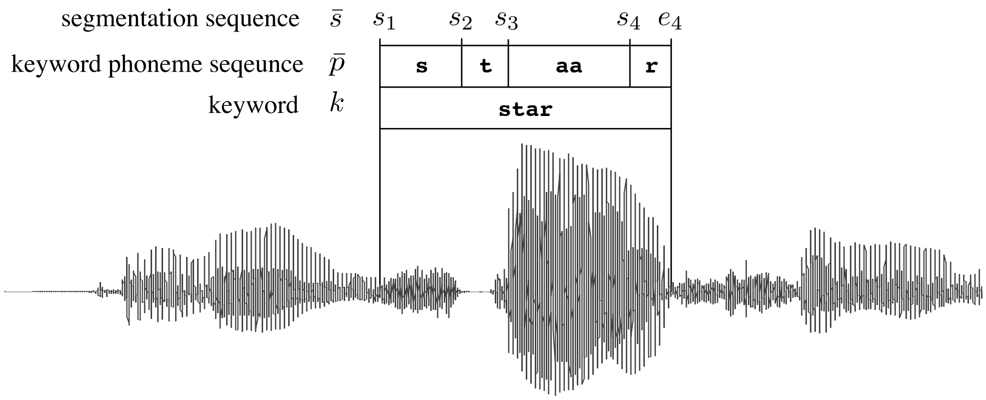

Fig. 1. Example of our notation. The waveform of the spoken utterance "a lone star shone..." taken from the TIMIT corpus. The keyword k is the word *star*. The phonetic transcription $\bar{p}$ along with its time span $\bar{s}$ are schematically depicted in the figure.

the paper we denote scalars using lower case Latin letters (e.g. x), and vectors using bold face letters (e.g. $\mathbf{x}$). A sequence of elements is designated by a bar ($\bar{\mathbf{x}}$) and its length is denoted as $|\bar{\mathbf{x}}|$.

Formally, we represent a speech signal as a sequence of acoustic feature vectors $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^d$ for all $1 \leq t \leq T$. We denote a keyword by $k \in \mathcal{K}$, where $\mathcal{K}$ is a lexicon of words. Each keyword k is composed of a sequence of phonemes $\bar{p}^k = (p_1, \ldots, p_L)$, where $p_l \in \mathcal{P}$ for all $1 \leq l \leq L$ and $\mathcal{P}$ is the domain of the phoneme symbols. We denote by $\mathcal{P}^*$ the set of all finite length sequences over $\mathcal{P}$. Let us further define the alignment between a phoneme sequence and a speech signal. We denote by $s_l \in \mathbb{N}$ the start time of phoneme p_l (in frame units), and by $e_l \in \mathbb{N}$ the end time of phoneme p_l. We assume that the start time of phoneme p_{l+1} is equal to the end time of phoneme p_l, that is, $e_l = s_{l+1}$ for all $1 \leq l \leq L-1$. The timing sequence (time span) $\bar{s}^k$ corresponding to the phoneme sequence $\bar{p}^k$ is a sequence of start-times and an end-time, $\bar{s}^k = (s_1, \ldots, s_L, e_L)$, where s_l is the start-time of phoneme p_l and e_L is the end-time of the last phoneme p_L. An example of our notation is given in Figure 1. Our goal is to learn a *keyword spotter*, denoted f, which takes as input the pair $(\bar{\mathbf{x}}, \bar{p}^k)$ and returns a real value expressing the confidence that the targeted keyword k is uttered in $\bar{\mathbf{x}}$. That is, f is a function from $\mathcal{X}^* \times \mathcal{P}^*$ to the set $\mathbb{R}$. The confidence score outputted by f for a given pair $(\bar{\mathbf{x}}, \bar{p}^k)$ can then be compared to a threshold $b \in \mathbb{R}$ to actually determine whether $\bar{p}^k$ is uttered in $\bar{\mathbf{x}}$.

The performance of a keyword spotting system is often measured by the Receiver Operating Characteristics (ROC) curve, that is, a plot of the true positive (spotting a keyword correctly) rate as a function of the false positive (mis-spotting a keyword) rate (see for example Benayed et al., 2004; Ketabdar et al., 2006; Silaghi and Bourlard, 1999). The points on the curve are obtained by sweeping the decision threshold b from the most positive confidence value outputted by the system to the most negative one. Hence, the choice of b represents a trade-off between different operational settings, corresponding to cost functions weighting false positive and false negative errors differently. Assuming a flat prior over all cost functions, it is appropriate to select the keyword spotting system maximizing the averaged performance over all settings, which corresponds to the model maximizing the area under the ROC curve (AUC). In the following we describe a large margin approach which aims at learning a keyword spotter which achieves high AUC.

3 A Large Margin Approach for Keyword Spotting

In this section we describe a discriminative algorithm for learning a spotting function f from a training set of examples. Our construction is based on a set of predefined feature functions $\{\phi_j\}_{j=1}^n$. Each feature function is of the form $\phi_j : \mathcal{X}^* \times \mathcal{P}^* \times \mathbb{N}^* \to \mathbb{R}$. That is, each feature function takes as input an acoustic representation of a speech utterance $\bar{\mathbf{x}} \in \mathcal{X}^*$, together with a phoneme sequence $\bar{p}^k \in \mathcal{P}^*$ of the keyword k, and a candidate time span $\bar{s} \in \mathbb{N}^*$, and returns a scalar in $\mathbb{R}$ which, intuitively, represents the confidence that the given keyword phoneme sequence is uttered in the suggested time span. For example, one feature function can sum the number of times phoneme p comes after phoneme p', while other feature function may extract properties of each acoustic feature vector $\mathbf{x}_t$ provided that phoneme p is pronounced at time t. The description of the concrete form of the feature functions is differed to Section 6.

Our goal is to learn a keyword spotter f, which takes as input a sequence of acoustic features $\bar{\mathbf{x}}$, a keyword $\bar{p}^k$, and returns a confidence value in $\mathbb{R}$. The form of the function f we use is

$$f(\bar{\mathbf{x}}, \bar{p}^k) = \max_{\bar{s}} \ \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}^k, \bar{s}) \ , \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^n$ is a vector of importance weights ("model parameters") that should be learned and $\boldsymbol{\phi} \in \mathbb{R}^n$ is a vector function composed out of the feature functions ϕ_j. In other words, f returns a confidence prediction about the existence of the keyword in the utterance by maximizing a weighted sum of the scores returned by the feature functions over all possible time spans. The maximization defined by Eq. (1) is over an exponentially large number of time spans. Nevertheless, as in HMMs, if the feature functions $\boldsymbol{\phi}$ are decomposable, the maximization in Eq. (1) can be efficiently calculated through dynamic programming as described in Section 5

Recall that we would like to obtain a system that attains high AUC on unseen data. In order to do so, we use two sets of training examples. Denote by $\mathcal{X}_k^+$ a set of speech utterances in which the keyword $k \in \mathcal{K}_{\text{train}}$ is uttered, where $\mathcal{K}_{\text{train}}$ is the set of all training keywords. Similarly, denote by $\mathcal{X}_k^-$ a set of speech utterances in which the keyword k is not uttered. The AUC for a keyword k can be written in the form of the *Wilcoxon-Mann-Whitney statistic* (Cortes and Mohri, 2004) as

$$A_k = \frac{1}{|\mathcal{X}_k^+||\mathcal{X}_k^-|} \sum_{\bar{\mathbf{x}}^+ \in \mathcal{X}_k^+} \sum_{\bar{\mathbf{x}}^- \in \mathcal{X}_k^-} \mathbb{1}_{\{f(\bar{\mathbf{x}}^+, \bar{p}^k) > f(\bar{\mathbf{x}}^-, \bar{p}^k)\}}, \tag{2}$$

where $|\cdot|$ refers to the cardinality of a set, and $\mathbb{1}_{\{\cdot\}}$ refers to the indicator function, that is, $\mathbb{1}_{\{\pi\}}$ is 1 whenever the predicate π is true and 0 otherwise. Thus, A_k estimates the probability that the score assigned to an utterance that contains the keyword k is greater than the score assigned to an utterance which does not contain it.

As one is often interested in the expected performance over any keyword, it is common to plot the ROC averaged over a set of evaluation keywords, $\mathcal{K}_{\text{test}}$, and to compute the corresponding averaged AUC,

$$A_{\text{test}} = \frac{1}{|\mathcal{K}_{\text{test}}|} \sum_{k \in \mathcal{K}_{\text{test}}} A_k \ . \tag{3}$$

In order to achieve the goal of high AUC, the proposed method picks a function f from the set of linear functions defined in Eq. (1), for which the following inequality holds: $f(\bar{\mathbf{x}}^+, \bar{p}^k) > f(\bar{\mathbf{x}}^-, \bar{p}^k)$, for every keyword $k \in \mathcal{K}_{\text{train}}$ and for as much utterance pairs $\mathbf{x}^+ \in \mathcal{X}_k^+$ and $\mathbf{x}^- \in \mathcal{X}_k^-$ as possible. Finding such a function is realized by learning the value of the weight vector $\mathbf{w}$ given a training set of examples.

We now describe a new approach based on large margin techniques for learning the weight vector $\mathbf{w}$ from a training set $\mathcal{S}$ of examples. Each example in the training set $\mathcal{S}$ is composed of a phoneme sequence $\bar{p}_i^{k_i}$ representing the keyword k_i, an utterance $\bar{\mathbf{x}}_i^+ \in \mathcal{X}_{k_i}^+$ in which the keyword k_i is uttered, an utterance $\bar{\mathbf{x}}_i^- \in \mathcal{X}_{k_i}^-$ in which the keyword k_i is not uttered, and the time span $\bar{s}_i^+$ of the phoneme sequence $\bar{p}_i^{k_i}$ in $\bar{\mathbf{x}}_i^+$. Overall we have m examples, that is, $\mathcal{S} = \{(\bar{p}_1^{k_1}, \bar{\mathbf{x}}_1^+, \bar{\mathbf{x}}_1^-, \bar{s}_1^+), \ldots, (\bar{p}_m^{k_m}, \bar{\mathbf{x}}_m^+, \bar{\mathbf{x}}_m^-, \bar{s}_m^+)\}$. Hence, we assume that we have access to the correct start times $\bar{s}_i^+$ of the phoneme sequence $\bar{p}_i^{k_i}$ in the positive training utterances $\bar{\mathbf{x}}_i^+ \in \mathcal{X}_{k_i}^+$ for all i. This assumption is actually not restrictive since such a timing sequence can be inferred by any forced-alignment algorithm, see (Keshet et al., 2007) and the reference therein. We evaluate the influence of forced-alignment compared to manual-alignment in Section 7.

For each keyword in the training set there is only one positive utterance and one negative utterance, hence $|\mathcal{X}_k^+| = 1$ and $|\mathcal{X}_k^-| = 1$ and $|\mathcal{K}_{\text{train}}| = m$, and the AUC of the training set becomes

$$A_{\text{train}} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\{f(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}) > f(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i})\}} \ . \tag{4}$$

Similarly to the SVM algorithm for binary classification (Cortes and Vapnik, 1995; Vapnik, 1998), our approach for choosing the weight vector $\mathbf{w}$ is based on the idea of large-margin separation. Theoretically, our approach can be described as a two-step procedure: first, we construct the vectors $\boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}, \bar{s}_i^+)$ and $\boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s})$ in the vector space $\mathbb{R}^n$ based on each instance $(\bar{p}_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+)$, and each possible time span $\bar{s}$ for the negative sequence $\bar{\mathbf{x}}_i^-$. Second, we find a vector $\mathbf{w} \in \mathbb{R}^n$, such that the projection of vectors onto $\mathbf{w}$ ranks the constructed vectors according to their quality. Ideally, for any keyword $k_i \in \mathcal{K}_{\text{train}}$, for every instance pair $(\bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-) \in \mathcal{X}_{k_i}^+ \times \mathcal{X}_{k_i}^-$, we would like the following constraint to hold

$$\mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}, \bar{s}_i^+) - \max_{\bar{s}} \ \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s}) \ \geq \ 1 \ \ \forall i \ . \tag{5}$$

That is, $\mathbf{w}$ should rank the utterance that contains the keyword above any utterance that does not contain it by at least 1. Moreover, we consider the best possible time span of the keyword within the utterance that does not contain it. We refer to the difference $\mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}, \bar{s}_i^+) - \max_{\bar{s}} \ \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s})$ as the *margin* of $\mathbf{w}$ with respect to the best time span of the keyword k_i in the utterance that does not contain it. Note that if the prediction of $\mathbf{w}$ is incorrect then the margin is negative. Naturally, if there exists a $\mathbf{w}$ satisfying all the constraints in Eq. (5), the margin requirements are also satisfied by multiplying $\mathbf{w}$ by a large scalar. The SVM algorithm solves this problem by selecting the weights $\mathbf{w}$ minimizing $\frac{1}{2}\|\mathbf{w}\|^2$ subject to the constraints given in Eq. (5), as it can be shown that the solution with the smallest norm is likely to achieve better generalization (Vapnik, 1998).

In practice, it might be the case that the constraints given in Eq. (5) cannot be satisfied. To overcome this obstacle, we follow the soft SVM approach (Cortes and Vapnik, 1995; Vapnik, 1998) and define the following hinge-loss function,

$$\ell(\mathbf{w}; (\bar{p}^k, \bar{\mathbf{x}}^+, \bar{\mathbf{x}}^-, \bar{s}^+)) = \Big[1 - \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}^+, \bar{p}^k, \bar{s}^+) + \max_{\bar{s}} \ \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}^-, \bar{p}^k, \bar{s})\Big]_+ \ , \tag{6}$$

where $[a]_+ = \max\{0, a\}$. The hinge loss measures the maximal violation for any of the constraints given in Eq. (5). The soft SVM approach for our problem is to choose the vector $\mathbf{w}^\star$ which minimizes the following optimization problem

$$\mathbf{w}^\star \;=\; \arg\min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \ell(\mathbf{w}; (\bar{p}_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+)) \quad , \tag{7}$$

where the parameter C serves as a complexity-accuracy trade-off parameter: a low value of C favors a simple model, while a large value of C favors a model which solves all training constraints (see Cristianini and Shawe-Taylor, 2000). Solving the optimization problem given in Eq. (7) is expensive since it involves a maximization for each training example. Most of the solvers for this problem, like SMO (Platt, 1998), iterate over the whole dataset several times until convergence. In the next section, we propose a slightly different method, which visits each example only once, and is based on our previous work (Crammer et al., 2006). Our method is shown to be competitive with the large margin approach and it is shown to attain high AUC over the training examples and over unseen examples (see Appendix A).

4 An Iterative Algorithm

We now describe a simple iterative algorithm for learning the weight vector $\mathbf{w}$. The algorithm receives as input a set of training examples $\mathcal{S} = \{(\bar{p}_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+)\}_{i=1}^m$ and examines each of them sequentially. Initially, we set $\mathbf{w} = \mathbf{0}$. At each iteration i, the algorithm updates $\mathbf{w}$ according to the current example $(\bar{p}_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+)$ as we now describe. Denote by $\mathbf{w}_{i-1}$ the value of the weight vector before the i-th iteration. Let $\bar{s}_i^-$ be the predicted time span for the negative utterance, $\bar{\mathbf{x}}_i^-$, according to $\mathbf{w}_{i-1}$,

$$\bar{s}_i^- = \arg\max_{\bar{s}} \;\mathbf{w}_{i-1}\cdot\boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s}) \;. \tag{8}$$

Let us define the difference between the feature functions of the acoustic sequence in which the keyword is uttered and the feature functions of the acoustic sequence in which the keyword is not uttered as $\Delta\boldsymbol{\phi}_i$, that is,

$$\Delta\boldsymbol{\phi}_i = \boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}, \bar{s}_i^+) - \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s}_i^-) \;. \tag{9}$$

We set the next weight vector $\mathbf{w}_i$ to be the minimizer of the following optimization problem,

$$\begin{aligned} &\min_{\mathbf{w}\in\mathbb{R}^n,\xi\geq 0} \quad \frac{1}{2}\|\mathbf{w}-\mathbf{w}_{i-1}\|^2 + C\;\xi \\ &\text{s.t.} \;\; \mathbf{w}\cdot\Delta\boldsymbol{\phi}_i \geq 1-\xi \;, \end{aligned} \tag{10}$$

where C serves as a complexity-accuracy trade-off parameter (see Crammer

Input: training set $\mathcal{S} = \{(\bar{p}_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+)\}_{i=1}^m$; validation set $\mathcal{S}_{\text{val}}$; parameter C

Initialize: $\mathbf{w}_0 = \mathbf{0}$

For $i = 1, \ldots, m$

Predict: $\bar{s}_i^- = \arg\max_{\bar{s}} \ \mathbf{w}_{i-1} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s})$

Set: $\Delta\boldsymbol{\phi}_i \ = \ \boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}, \bar{s}_i^+) - \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s}_i^-)$

If $\mathbf{w}_{i-1} \cdot \Delta\boldsymbol{\phi}_i < 1$

Set: $\alpha_i = \min\left\{C \ , \ \dfrac{1 - \mathbf{w}_{i-1} \cdot \Delta\boldsymbol{\phi}_i}{\|\Delta\boldsymbol{\phi}_i\|^2}\right\}$

Update: $\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \cdot \Delta\boldsymbol{\phi}_i$

Output: The weight vector $\mathbf{w}^*$ which achieves best AUC performance on the validation set $\mathcal{S}_{\text{val}}$ of size m_{val}:

$$\mathbf{w} = \mathop{\arg\min}_{\mathbf{w} \in \{\mathbf{w}_1, \ldots, \mathbf{w}_m\}} \frac{1}{m_{\text{val}}} \sum_{j=1}^{m_{\text{val}}} \mathbb{1}_{\{\max_{\bar{s}^+} \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_j^+, \bar{p}_j^{k_j}, \bar{s}^+) \ > \ \max_{\bar{s}^-} \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_j^-, \bar{p}_j^{k_j}, \bar{s}^-)\}}$$

Fig. 2. An iterative algorithm.

et al., 2006) and ξ is a non-negative slack variable, which indicates the loss of the ith example. Intuitively, we would like to minimize the loss of the current example, i.e., the slack variable ξ, while keeping the weight vector $\mathbf{w}$ as close as possible to the previous weight vector $\mathbf{w}_{i-1}$. The constraint makes the projection of the sequence that contains the keyword onto $\mathbf{w}$ higher than the projection of the sequence that does not contain it onto $\mathbf{w}$ by at least 1. It can be shown (see Crammer et al., 2006) that the solution to the above optimization problem is

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \Delta\boldsymbol{\phi}_i \ . \tag{11}$$

The value of the scalar α_i is based on the difference $\Delta\boldsymbol{\phi}_i$, the previous weight vector $\mathbf{w}_{i-1}$, and a parameter C. Formally,

$$\alpha_i = \min\left\{C, \ \frac{[1 - \mathbf{w}_{i-1} \cdot \Delta\boldsymbol{\phi}_i]_+}{\|\Delta\boldsymbol{\phi}_i\|^2}\right\} \quad . \tag{12}$$

The optimization problem given in Eq. (10) is based on recent work on online learning algorithms (Crammer et al., 2006). Based on this work, it is shown in Appendix A that, under some mild technical conditions, the cumulative AUC of the iterative procedure, i.e., $\frac{1}{m}\sum_{i=1}^m \mathbb{1}_{\{\mathbf{w}_i \cdot \Delta\boldsymbol{\phi}_i > 0\}}$ is likely to be high. Moreover, the appendix further shows that given the high cumulative AUC, there exists at least one weight vector among the vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$ which attains high averaged AUC on unseen examples as well. To find such a weight vector, we simply calculate the averaged loss attained by each of the weight vectors on a validation set. A pseudo-code of our algorithm is given in Figure 2.

In the case the user would like to select a threshold b that would ensure a specific requirement in terms of true positive rate or false negative rate, a simple cross-validation procedure (see Bengio et al., 2005) would consist in selecting the confidence value given by our model at the point of interest over the ROC curve plotted for some validation utterances of the targeted keyword.

5 Efficiency and Complexity

We now describe the problem of efficient evaluation of the function f given in Eq. (1). the evaluation of f requires solving the following optimization problem,

$$f(\bar{\mathbf{x}}, \bar{p}^k) = \max_{\bar{s}} \ \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}^k, \bar{s}) \ \ .$$

Similarly, we need to find an efficient way for solving the maximization problem given in Eq. (8). A direct search for the maximizer is not feasible since the number of possible time spans, $\bar{s}$, is exponential in the number of phonemes. Fortunately, as we show below, by imposing a few mild conditions on the structure of the feature functions both problems can be solved in polynomial time.

For simplicity, we assume that each feature function, ϕ_j, can be decomposed as follows. Let $\hat{\phi}_j$ be any function from $\mathcal{X}^* \times \mathcal{P}^* \times \mathbb{N}^3$ into the reals, which can be computed in a constant time. That is, $\hat{\phi}_j$ receives as input the signal, $\bar{\mathbf{x}}$, the sequence of phonemes, $\bar{p}^k$, and three time points. Additionally, we use the convention $s_0 = 0$ and $s_{|\bar{p}^k|+1} = T + 1$. Using the above notation, we assume that each ϕ_j can be decomposed to be

$$\phi_j(\bar{\mathbf{x}}, \bar{p}^k, \bar{s}) \ = \ \sum_{l=2}^{|\bar{s}|-1} \hat{\phi}_j(\bar{\mathbf{x}}, \bar{p}^k, s_{l-1}, s_l, s_{l+1}) \ \ . \tag{13}$$

The feature functions we describe in the next section can be decomposed as in Eq. (13).

We now describe an efficient algorithm for calculating the best time span assuming that ϕ_j can be decomposed as in Eq. (13). Given phoneme index $l \in \{1, \ldots, |\bar{p}^k|\}$ and two time indices $t, t' \in \{1, \ldots, T\}$, denote by $D(l, t, t')$ the score for the prefix of the phoneme index sequence $1, \ldots, l$, assuming that their actual start times are $s_1, \ldots, s_l$, where $s_l = t'$ and assuming that $s_{l+1} = t$. This variable can be computed efficiently in a similar fashion to the forward variables calculated by the Viterbi procedure in HMMs (see for

Input: speech signal $\bar{\mathbf{x}}$; sequence of phonemes $\bar{p}^k$; weight vector $\mathbf{w}$; maximum phoneme duration $L_{\max}$; minimum phoneme duration $L_{\min}$

Initialize: $\forall(1 \le t \le L_{\max}),\ D(0,t,0) = 0$

Recursion:

For $s = 1, \ldots, T$

For $l = 1, \ldots, |\bar{p}^k|$

For $t = s + lL_{\min}, \ldots, s + lL_{\max}$

For $t' = t - L_{\max}, \ldots, t - L_{\min}$

$$D(l,t,t') = \max_{t'-L_{\max} \le t'' < t'-L_{\min}} D(l-1,t',t'') + \mathbf{w} \cdot \hat{\boldsymbol{\phi}}(\bar{\mathbf{x}}, \bar{p}^k, t'', t', t)$$

$$D^\star_s = \max_{s+|\bar{p}^k|L_{\min} \le t \le s+|\bar{p}^k|L_{\max}} \ \max_{t-L_{\max} \le t' \le t-L_{\min}} D(|\bar{p}^k|, t, t')$$

Termination: $D^\star = \max_s D^\star_s, \quad \bar{s}^\star = \arg\max_s D^\star_s$

Fig. 3. An efficient procedure for evaluating the keyword spotting function.

instance Rabiner and Juang, 1993). The pseudo code for computing $D(l,t,t')$ recursively is shown in Figure 3. The best time span, $\bar{s}^\star$, is obtained from the algorithm by saving the intermediate values that maximize each expression in the recursion step. The complexity of the decoding is $\mathcal{O}(|\bar{p}^k|\ |\bar{\mathbf{x}}|^4)$. However, in practice, we can use the assumption that the maximal length of a phoneme is bounded, $t - t' \le L_{\max}$. This assumption reduces the complexity of the decoding down to $\mathcal{O}(|\bar{p}^k|\ |\bar{\mathbf{x}}|\ L^3_{\max})$. For comparison, the complexity of the decoding in standard Viterbi-based HMM is $\mathcal{O}(|P + \bar{p}^k|\ |\bar{\mathbf{x}}|)$.

To conclude this section we discuss the global complexity of our proposed method. In the training phase, our algorithm performs m iterations, one iteration per training example. At each iteration the algorithm evaluates the keyword spotting function once, updates the keyword spotting function, if needed, and evaluates the new function on a validation set of size m_{val}. Each evaluation of the function takes an order of $\mathcal{O}(|\bar{p}^k|\ |\bar{\mathbf{x}}|\ L^3_{\max})$ operations. Therefore the total complexity of our method becomes $\mathcal{O}(m\ m_{\text{val}}\ |\bar{p}^k|\ |\bar{\mathbf{x}}|\ L^3_{\max})$. In practice, however, we can evaluate the updated keyword spotting function only for the last 20 iterations or so, which reduces the global complexity of the algorithm to $\mathcal{O}(m\ |\bar{p}^k|\ |\bar{\mathbf{x}}|\ L^3_{\max})$. In all of our experiments, evaluating the keyword spotting function only for the last 20 iterations was found empirically to give sufficient results.

6 Feature Functions

In this section we present the implementation details of our learning approach for the task of keyword spotting. Recall that our construction is based on a set of feature functions, $\{\phi_j\}_{j=1}^n$, which maps an acoustic-phonetic representation of a speech utterance as well as a suggested time span of the keyword into a vector-space. In order to make this section more readable we omit the keyword index k.

We introduce a specific set of base functions, which is highly adequate for the keyword spotting problem. We utilize seven different feature functions ($n = 7$). These feature functions are used for defining our keyword spotting function $f(\bar{\mathbf{x}}, \bar{p})$ as in Eq. (1). Note that the same set of feature functions is also useful in the task of large-margin forced-alignment (Keshet et al., 2007), and they are given here only for completeness. A detailed analysis of this feature set is given in (Keshet et al., 2007).

Our first four feature functions aim at capturing transitions between phonemes. These feature functions are the distance between frames of the acoustic signal at both sides of phoneme boundaries as suggested by a timing sequence $\bar{s}$. The distance measure we employ, denoted by d, is the Euclidean distance between feature vectors. Our underlying assumption is that if two frames, $\mathbf{x}_t$ and $\mathbf{x}_{t'}$, are derived from the same phoneme then the distance $d(\mathbf{x}_t, \mathbf{x}_{t'})$ should be smaller than if the two frames are derived from different phonemes. Formally, our first four feature functions are defined as

$$\phi_j(\bar{\mathbf{x}}, \bar{p}, \bar{s}) = \frac{1}{|\bar{p}|} \sum_{l=2}^{|\bar{p}|-1} d(\mathbf{x}_{-j+s_l}, \mathbf{x}_{j+s_l}), \quad j \in \{1, 2, 3, 4\} \ . \tag{14}$$

If $\bar{s}$ is the correct time span then distances between frames across the phoneme change points are likely to be large. In contrast, an incorrect phoneme start time sequence is likely to compare frames from the same phoneme, often resulting in small distances.

The fifth feature function we use is built from kernel-based frame-wise phoneme classifier described in Dekel et al. (2004). Formally, for each phoneme event $p \in \mathcal{P}$ and frame $\mathbf{x} \in \mathcal{X}$, there is a confidence, denoted $g_p(\mathbf{x})$, that the phoneme p is pronounced in the frame $\mathbf{x}$. The resulting feature function measures the cumulative confidence of the complete speech signal given the phoneme sequence and their start-times,

$$\phi_5(\bar{\mathbf{x}}, \bar{p}, \bar{s}) = \frac{1}{|\bar{p}|} \sum_{l=1}^{|\bar{p}|} \frac{1}{s_{l+1} - s_l} \sum_{t=s_l}^{s_{l+1}-1} g_{p_l}(\mathbf{x}_t) \ . \tag{15}$$

Our next feature function scores timing sequences based on phoneme durations. Unlike the previous feature functions, the sixth feature function is oblivious to the speech signal itself. It merely examines the length of each phoneme, as suggested by $\bar{s}$, compared to the typical length required to pronounce this phoneme. Formally,

$$\phi_6(\bar{\mathbf{x}}, \bar{p}, \bar{s}) = \frac{1}{|\bar{p}|} \sum_{l=1}^{|\bar{p}|} \log \ \mathcal{N}(s_{l+1} - s_l; \hat{\mu}_{p_l} \ , \hat{\sigma}_{p_l}) \quad , \tag{16}$$

where $\mathcal{N}$ is a Normal probability density function with mean $\hat{\mu}_p$ and standard deviation $\hat{\sigma}_p$. In our experiments, we estimated $\hat{\mu}_p$ and $\hat{\sigma}_p$ from the training set (see Section 7).

Our last feature function exploits assumptions on the speaking rate of a speaker. Intuitively, people usually speak in an almost steady rate and therefore a timing sequence in which speech rate is changed abruptly is probably incorrect. Formally, let $\hat{\mu}_p$ be the average length required to pronounce the pth phoneme. We denote by r_l the relative speech rate, $r_l = (s_{l+1} - s_l)/\hat{\mu}_{p_l}$. That is, r_l is the ratio between the actual length of phoneme p_l as suggested by $\bar{s}$ to its average length. The relative speech rate presumably changes slowly over time. In practice the speaking rate ratios often differ from speaker to speaker and within a given utterance. We measure the local change in the speaking rate as $(r_l - r_{l-1})^2$ and we define the feature function ϕ_7 as the local change in the speaking rate,

$$\phi_7(\bar{\mathbf{x}}, \bar{p}, \bar{s}) = \frac{1}{|\bar{p}|} \sum_{l=2}^{|\bar{p}|} (r_l - r_{l-1})^2 \quad . \tag{17}$$

7 Experimental Results

In this section we present experimental results that demonstrate the robustness of our proposed discriminative system compared to context-independent HMM-based system. We performed experiments on read speech using the TIMIT, HTIMIT and WSJ corpora and on spontaneous speech using the OGI Stories corpus. In all the experiments, the baseline discriminative system and HMM system were trained on the clean read-speech TIMIT corpus. We divided the training portion of TIMIT (excluding the SA1 and SA2 utterances) into two disjoint parts containing 500, and 3196 utterances. The first part of the training set was used for learning the functions g_p, which define the feature function ϕ_5, according to Eq. (15). These functions were learned by the kernel-based phoneme classification algorithm described in Dekel et al. (2004) using the MFCC+Δ+$\Delta\Delta$ acoustic features and a Gaussian kernel with parameter $\sigma = 6.24$ (selected on the validation set). Using the functions g_p

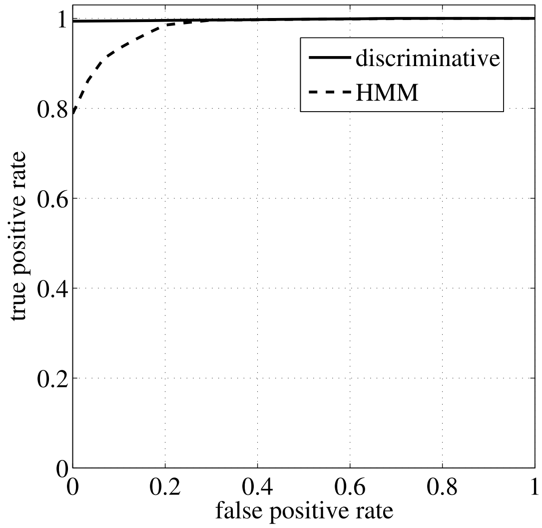

Fig. 4. ROC curves of the models generated from the discriminative system and the context-independent HMM-based system, both trained on the TIMIT training set and evaluated on 80 keywords from TIMIT test set. The AUC are **0.995** and **0.941** for the discriminative system and the HMM-based system, respectively.

as a frame-based phoneme classifier resulted in classification accuracy of 55% per frame on the TIMIT core test set.

The second set of 3196 utterances formed the training set for the keyword spotter. From this set we picked 200 random keywords for training and 200 different keywords for validation. The keywords were chosen to have a minimum length of at least 6 phonemes. For each of the keywords we chose one positive utterance in which the keyword was uttered and one negative utterance in which the keyword was not uttered. The same utterance could be a positive utterance for one keyword and a negative utterance for a different keyword, but in any case the utterances used for the training were not used for validation. The iterative discriminative algorithm was very robust to the set of training and validation keyword set and picking a different set led to similar performance results. We ran the iterative discriminative algorithm with value of the parameter $C = 1$. Overall the algorithm has to estimate a total of 7 parameters, along with the phoneme classification described above.

Since the proposed discriminative system is implemented with context-independent feature functions, we compared it to a context-independent HMM-based system. We trained a context-independent HMM phoneme recognizer from the entire TIMIT training portion, where 3600 utterances were used as a training set and 96 utterances were used as a validation set. In our setting each phoneme was represented by a simple left-to-right HMM of 5 emitting states with 40 diagonal Gaussians. These models were enrolled as follows: first

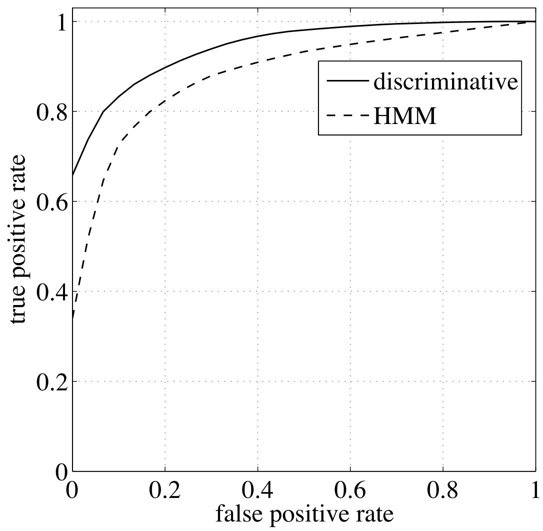

Fig. 5. ROC curves of the models generated from the discriminative system and the context-independent HMM-based system, both trained on the TIMIT training set and evaluated on 80 keywords from WSJ test set. The AUC are **0.942** and **0.88** for the discriminative system and the HMM-based system, respectively.

the HMMs were initialized using K-means, and then enrolled independently using EM and the segmentation provided by TIMIT. The second step, often called *embedded training*, re-enrolls all the models by relaxing the segmentation constraints using a forced-alignment. Minimum values of the variances for each Gaussian were set to 20% of the global variance of the data. All HMM experiments were done using the *Torch* package (Collobert et al., 2002). All hyper-parameters including the number of states, the number of Gaussians per state, and the variance flooring factor, were tuned using the validation set. The number of parameters in the HMM model can be calculated as follows. There are 5 states per phone, 40 Gaussians per states, 39 phones, and the data is 39 dimensional; hence there are $(40 + 2 \times 40 \times 39) \times 5 \times 39$ parameters for the emission distributions and $(8 \times 39) + (39 \times 39)$ parameters for the transition distributions. Overall there are $618,033$ parameters in the HMM system. The resulting HMM was a context-independent state-of-the-art phoneme recognizer with accuracy of 64% on the TIMIT test set.

Keyword detection using the HMM-based system was performed with a HMM composed of two context-independent sub-HMMs, the keyword model and the garbage model, as depicted in Figure 6. The keyword model was a context-independent HMM which estimated the likelihood of an acoustic sequence given that the sequence represented the keyword phoneme sequence. The garbage model was an HMM composed of context-independent phoneme HMMs fully connected to each others, which estimated the likelihood of any acoustic sequence. The overall HMM fully connected the keyword model and

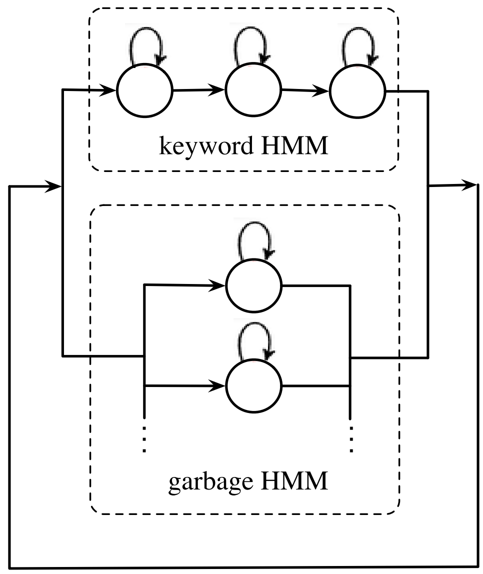

Fig. 6. HMM topology for keyword spotting.

the garbage model. The detection of a keyword given a test utterance was performed through a best path search, where an external parameter of the prior keyword probability was added to the keyword sub-HMM. The best path found by Viterbi decoding on the overall HMM either passed through the keyword model (in which case the keyword was said to be uttered) or not (in which case the keyword was assumed not to be in the acoustic sequence). Swiping the prior keyword probability parameters set the trade-off between the true positive rate and the false positive rate.

The test set was composed of 80 randomly chosen keywords, distinct from the keywords of the training and validation sets (the list of keyword for this experiment as well as for all other experiments is given in Appendix B). The keywords were selected from the TIMIT dictionary to have a minimal length of 4 phonemes. For each keyword, we randomly picked at most 20 utterances in which the keyword was uttered and at most 20 utterances in which it was not uttered. Note that the number of test utterances in which the keyword was uttered was not always 20, since some keywords were uttered less than 20 times in the whole TIMIT test set. Both the discriminative system and the HMM-based system were evaluated against the test data. The results are reported as averaged ROC curves in Figure 4. The AUC are 0.995 and 0.941 for the discriminative system and the context-independent HMM-based system, respectively. In order to check whether the advantage over the averaged AUC could be due to a few keyword, we ran the Wilcoxon test. At the 95% confidence level, the test rejected this hypothesis, showing that the discriminative system indeed brings a consistent improvement on the keyword set.

In order to make sure that the our learning procedure of the proposed dis-

Table 1
Comparison of training the discriminative system with the TIMIT manual phoneme alignment and with automatic forced-alignment. The AUC of the discriminative system on the TIMIT test set for both cases is compared to the context-independent HMM-based system.

Training alignment	Discriminative test set AUC	HMM test set AUC
TIMIT manual alignment	**0.995**	0.941
TIMIT forced alignment	**0.996**	0.941

criminative algorithm can be applied in absence of manual alignment data, we trained a model on phoneme time spans extracted from forced aligned data. We used the algorithm presented in (Keshet et al., 2007) for forced alignment, where it was trained on 50 utterances of TIMIT training portion which was not used for training or validation the keyword spotting algorithm. The AUC of the resulted discriminative system trained with forced aligned data and evaluated on the TIMIT test data described above was 0.996, almost identical to the AUC of the discriminative system trained on manual aligned data. The results are given in Table 1.

In the next experiments we examine the robustness of the proposed discriminative system to different environments. We used the discriminative system and the HMM-based systems trained on TIMIT and evaluated them on different corpora without any further training or adaptation. For the discriminative system, we used the manually aligned trained model, since there was no significant difference between the manually aligned and the forced aligned models. First we evaluated the systems on the HTIMIT corpus (Reynolds, 1997). The HTIMIT corpus was generated by playing the TIMIT speech through a loudspeaker into a different set of phone handsets. The TIMIT trained systems were tested on a set of 80 keywords which were not used in the training set. For each keyword, we randomly picked at most 20 utterances in which the keyword was uttered and at most 20 utterances in which it was not uttered from the CB1 portion of the HTIMIT corpus. The AUC are 0.949 and 0.922 for the discriminative system and the context-independent HMM-based system, respectively. With more than 99% confidence, the Wilcoxon test rejected the hypothesis that the difference between the two systems was due to only a few keywords. Hence, these experiments on HTIMIT show that the introduction of channel variations degrades the performance of both systems, but does not change the relative advantage of the discriminative system over the HMM-based system.

Next, we compared the performance of the systems on the Wall Street Journal (WSJ) corpus (Paul and Baker, 1992). This corpus corresponds to read

Table 2
Summary of the empirical performance of the discriminative system and the context-independent HMM-based system in all experiments.

Corpus	Discriminative AUC	HMM AUC
TIMIT	**0.996**	0.941
HTIMIT	**0.949**	0.922
WSJ	**0.942**	0.87
OGI Stories	**0.769**	0.722

articles of the Wall Street Journal, and hence presents a different linguistic context compared to TIMIT. Both the discriminative system and the context-independent HMM-based system were trained on the TIMIT corpus as described above and evaluated on a different set of 80 keywords from the WSJ corpus. For each keyword, we randomly picked at most 20 utterances in which the keyword was uttered and at most 20 utterances in which it was not uttered from the `si_tr_s` portion of the WSJ corpus. The ROC curves are given in Figure 5. The AUC are 0.942 and 0.88 for the discriminative model and the context-independent HMM-based system, respectively. With more than 99% confidence, the Wilcoxon test rejected the hypothesis that the difference between the two systems was due to only a few keywords.

Last, we compared the performance of the systems on OGI Stories corpus [1]. In this corpus, spontaneous speech was recorded by asking American speakers to talk freely about a topic of their choice. Again, both systems were trained on the TIMIT corpus as described above and evaluated on a different set of 60 keywords. For each keyword, we randomly picked at most 20 utterances in which the keyword was uttered and at most 20 utterances in which it was not uttered. The results of these experiments on spontaneous speech are consistent with the results obtained on read speech. Indeed, the discriminative system outperforms the HMM-based system, with an AUC of 0.769 compared to 0.722, respectively. As in previous cases, the Wilcoxon test rejected the hypothesis that the difference between the two systems was due to only a few keywords, at the 95% confidence level.

A summary of the results of all experiments is given in Table 2. A closer look on them shows that the discriminative system systematically outperforms the context-independent HMM-based system in terms of AUC. This indeed validates our hypothesis that it is a good strategy to maximize the AUC. Moreover, the discriminative system outperforms the HMM-based system for

[1] `http://cslu.cse.ogi.edu/corpora/stories/`

all points of the ROC curve, meaning that it has better true positive rate for every given false negative rate.

Finally, we would like to note that a complete description of the experimental setup, the keywords and the utterances used for evaluating and a source code in C++ of the keyword spotter can be found in `http://www.idiap.ch/keyword_spotting_benchmark/`.

8 Summary

Keyword spotting is a speech related task with more and more practical interest from an application point of view. Current state-of-the-art approaches are based on classical generative HMM-based systems. In this work, we introduced a discriminative approach to keyword spotting, aimed at attaining high area under the ROC curve, i.e., the most common measure for keyword spotter evaluation. Furthermore, the proposed approach is based on a large-margin formulation of the problem (hence expecting a good generalization performance) and an iterative training algorithm (hence expecting to scale reasonably well to large databases). Compared to conventional context-independent HMM-based system, the proposed discriminative system has shown to yield a statistically significant improvement on the TIMIT corpus. Furthermore, the very same system trained on the TIMIT corpus was tested on different corpora to assess its performance in various conditions. Namely, the system has been assessed on HTIMIT, which introduces various channel variations, on WSJ, which introduces different types of sentences from a linguistic perspective, and on OGI Stories, which corresponds to the recording of spontaneous speech. In all cases, the discriminative system was shown to yield a statistically better performance than the context-independent HMM alternative.

We would like to note that this work is part of a general line of research on large margin and kernel method for discriminative continuous speech recognition. Dekel et al. (2004) described and analyzed an hierarchical approach for frame-based phoneme classification. Building on that work, we proposed an large-margin based discriminative algorithm for forced-alignment (Keshet et al., 2007), and an algorithm for whole sequence phoneme recognition (Keshet et al., 2006). The discriminative keyword spotting presented in this paper is in turn based on those works and it is the first to address word-level recognition. We are currently investigating an extension of this work to large-margin discriminative large vocabulary continuous speech recognition. We are also looking for a method to encompass contextual information into our models. Last but not least, we are working on reducing the heavy computational load required by kernel-based algorithms, with the objective to reach the efficiency of HMM-based solutions.

Last, a complete description of the experiments along with a source code for the discriminative keyword spotter can be found in `http://www.idiap.ch/keyword_spotting_benchmark/`.

Acknowledgement. Part of this work was supported by EU project DIRAC (FP6-0027787). Most of this work has been performed while David Grangier was with the IDIAP Research Institute. The authors wish to thank the anonymous reviewers for their helpful comments, which have enhanced this paper.

A Theoretical Analysis

In this appendix, we show that the iterative algorithm given in Section 4 attains high cumulative AUC, defined as

$$\hat{A}_{\text{train}} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\{\mathbf{w}_i \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}, \bar{s}_i^+) \geq \mathbf{w}_i \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s}_i^-)\}}, \tag{A.1}$$

where $\bar{s}_i^-$ is predicted every iteration step according to Eq. (8). The examination of the cumulative AUC is of great interest as it provides an estimator for the generalization performance. Note that at each iteration step the iterative algorithm receives new example $(p_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+)$ and predicts the time span of the keyword in the negative instance $\bar{\mathbf{x}}_i^-$ using the previous weight vector $\mathbf{w}_{i-1}$. Only after the prediction is made the algorithm suffers loss by comparing its prediction to the true time span $\bar{s}_i^+$ of the keyword on the positive utterance $\bar{\mathbf{x}}_i^+$. The cumulative AUC is a weighted sum of the performance of the algorithm on the next unseen training example and hence it is a good estimation to the performance of the algorithm on unseen data during training.

Our first theorem states a competitive bound. It compares the cumulative AUC of the weight vectors series, $\{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$, resulted from the iterative algorithm to the best fixed weight vector, $\mathbf{w}^\star$, chosen in hindsight, and essentially proves that, for any sequence of examples, our algorithms cannot do much worse than the best fixed weight vector. Formally, it shows that the cumulative area *above* the curve, $1 - \hat{A}_{\text{train}}$, is smaller than the weighted average loss $\ell(\mathbf{w}^\star; (\bar{p}_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+))$ of the best fixed weight vector $\mathbf{w}^\star$ and its weighted complexity, $\|\mathbf{w}^\star\|$. That is, the cumulative AUC of the iterative training algorithm is going to be high, given that the loss of the best solution is small, the complexity of the best solution is small and that the number of training examples, m, is sufficiently large.

Theorem 1 *Let $\mathcal{S} = \{(\bar{p}_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+)\}_{i=1}^m$ be a set of training examples and assume that for all k, $\bar{\mathbf{x}}$ and $\bar{s}$ we have that $\|\boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}^k, \bar{s})\| \leq 1/\sqrt{2}$. Let $\mathbf{w}^\star$ be* *the best weight vector selected under some optimization criterion by observing all instances in hindsight. Let $\mathbf{w}_1, \ldots, \mathbf{w}_m$ be the sequence of weight vectors obtained by the algorithm in Figure 2 given the training set $\mathcal{S}$. Then,*

$$1 - \hat{A}_{\text{train}} \leq \frac{1}{m}\|\mathbf{w}^\star\|^2 + \frac{2C}{m}\sum_{i=1}^{m} \ell(\mathbf{w}^\star; (\bar{p}_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+)). \tag{A.2}$$

where $C \geq 1$ and $\hat{A}_{\text{train}}$ is the cumulative AUC defined in Equation A.1.

Proof Denote by $\ell_i(\mathbf{w})$ the instantaneous loss the weight vector $\mathbf{w}$ suffers on the i-th example, that is,

$$\ell_i(\mathbf{w}) = [1 - \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}, \bar{s}_i^+) + \max_{\bar{s}} \ \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s})]_+$$

The proof of the theorem relies on Lemma 1 and Theorem 4 in Crammer et al. (2006). Lemma 1 in Crammer et al. (2006) implies that,

$$\sum_{i=1}^{m} \alpha_i \left(2\ell_i(\mathbf{w}_{i-1}) - \alpha_i\|\Delta\boldsymbol{\phi}_i\|^2 - 2\ell_i(\mathbf{w}^\star)\right) \leq \|\mathbf{w}^\star\|^2. \tag{A.3}$$

Now if the algorithm makes a prediction mistake and the predicted confidence of the best time span of the keyword in a negative utterance is higher than the confidence of the true time span of the keyword in the positive example then $\ell_i(\mathbf{w}_{i-1}) \geq 1$. Using the assumption that $\|\boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}^k, \bar{s})\| \leq 1/\sqrt{2}$, which means that $\|\Delta\boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}^k, \bar{s})\|^2 \leq 1$, and the definition of α_i given in Eq. (12), when substituting $[1 - \mathbf{w}_{i-1} \cdot \Delta\boldsymbol{\phi}_i]_+$ for $\ell_i(\mathbf{w}_{i-1})$ in its numerator, we conclude that if a prediction mistake occurs then it holds that

$$\alpha_i \ell_i(\mathbf{w}_{i-1}) \geq \min\left\{\frac{\ell_i(\mathbf{w}_{i-1})}{\|\Delta\boldsymbol{\phi}_i\|^2}, C\right\} \geq \min\{1, C\} = 1. \tag{A.4}$$

Summing over all the prediction mistakes made on the entire training set $\mathcal{S}$ and taking into account that $\alpha_i\ell_i(\mathbf{w}_{i-1})$ is always non-negative, we have

$$\sum_{i=1}^{m} \alpha_i \ell_i(\mathbf{w}_{i-1}) \geq \sum_{i=1}^{m} \mathbb{1}_{\{\mathbf{w}_{i-1}\cdot\boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}, \bar{s}_i^+) \leq \mathbf{w}_{i-1}\cdot\boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s}_i^-)\}}. \tag{A.5}$$

Again using the definition of α_i, we know that $\alpha_i\ell_i(\mathbf{w}^\star) \leq C\ell_i(\mathbf{w}^\star)$ and that $\alpha_i\|\Delta\boldsymbol{\phi}_i\|^2 \leq \ell_i(\mathbf{w}_{i-1})$. Plugging these two inequalities and Eq. (A.5) into Eq. (A.3) we get

$$\sum_{i=1}^{m} \mathbb{1}_{\{\mathbf{w}_{i-1}\cdot\boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}_i^{k_i}, \bar{s}_i^+) \leq \mathbf{w}_{i-1}\cdot\boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}_i^{k_i}, \bar{s}_i^-)\}} \leq \|\mathbf{w}^\star\|^2 + 2C\sum_{i=1}^{m} \ell_i(\mathbf{w}^\star). \tag{A.6}$$

The theorem follows by replacing the sum over prediction mistakes to a sum over prediction hits and plugging the definition of the cumulative AUC given in Eq. (A.1). $\square$

The next theorem states that the output of our algorithm is likely to have good generalization, i.e., the expected value of the AUC resulted from decoding on unseen test set is likely to be large.

Theorem 2 *Under the same conditions of Theorem 1. Assume that the training set $\mathcal{S}$ and the validation set $\mathcal{S}_{\mathrm{val}}$ are both sampled i.i.d. from a distribution Q. Denote by m_{val} the size of the validation set. With probability of at least $1-\delta$ we have*

$$1 - A_{\mathrm{test}} = \mathbb{E}_Q\left[\mathbb{1}_{\{f(\bar{\mathbf{x}}_i^+,\bar{p}_i^{k_i}) \le f(\bar{\mathbf{x}}_i^-,\bar{p}_i^{k_i})\}}\right] = \Pr_Q\left[f(\bar{\mathbf{x}}_i^+,\bar{p}_i^{k_i}) \le f(\bar{\mathbf{x}}_i^-,\bar{p}_i^{k_i})\right] \le$$

$$\frac{1}{m}\sum_{i=1}^{m} \ell(\mathbf{w}^\star; (\bar{p}_i^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^+)) + \frac{\|\mathbf{w}^\star\|^2}{m} + \frac{\sqrt{2\ln(2/\delta)}}{\sqrt{m}} + \frac{\sqrt{2\ln(2m/\delta)}}{\sqrt{m_{\mathrm{val}}}} \quad , \qquad \text{(A.7)}$$

where A_{test} is the mean AUC defined as $A_{\mathrm{test}} = \mathbb{E}_Q\left[\mathbb{1}_{\{f(\bar{\mathbf{x}}_i^+,\bar{p}_i^{k_i}) > f(\bar{\mathbf{x}}_i^-,\bar{p}_i^{k_i})\}}\right]$.

Proof Denote the risk of keyword spotter f by

$$\mathrm{risk}(f) = \mathbb{E}\left[\mathbb{1}_{\{f(\bar{\mathbf{x}}_i^+,\bar{p}_i^{k_i}) \le f(\bar{\mathbf{x}}_i^-,\bar{p}_i^{k_i})\}}\right] = \Pr\left[f(\bar{\mathbf{x}}_i^+,\bar{p}_i^{k_i}) \le f(\bar{\mathbf{x}}_i^-,\bar{p}_i^{k_i})\right]$$

Proposition 1 in (Cesa-Bianchi et al., 2004) implies that with probability of at least $1-\delta_1$ the following bound holds,

$$\frac{1}{m}\sum_{i=1}^{m} \mathrm{risk}(f_i) \;\le\; \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{\{f_i(\bar{\mathbf{x}}_i^+,\bar{p}_i^{k_i}) \le f_i(\bar{\mathbf{x}}_i^-,\bar{p}_i^{k_i})\}} + \frac{\sqrt{2\ln(1/\delta_1)}}{\sqrt{m}} \quad .$$

Combining this fact with Theorem 1 we obtain that,

$$\frac{1}{m}\sum_{i=1}^{m} \mathrm{risk}(f_i) \;\le\; \frac{2C}{m}\sum_{i=1}^{m} \ell_i(\mathbf{w}^\star) + \frac{\|\mathbf{w}^\star\|^2}{m} + \frac{\sqrt{2\ln\left(1/\delta_1\right)}}{\sqrt{m}} \quad . \qquad \text{(A.8)}$$

The left-hand side of the above inequality upper bounds $\mathrm{risk}(f^\star)$, where $f^\star = \arg\min_{f_i} \mathrm{risk}(f_i)$. Therefore, among the finite set of keyword spotting functions, $\mathcal{F} = \{f_1, \ldots, f_m\}$, there exists at least one keyword spotting function (for instance the function $f^\star$) whose true risk is bounded above by the right hand side of Eq. (A.8). Recall that the output of our algorithm is the keyword spotter $f \in \mathcal{F}$, which maximizes the average AUC over the validation set $\mathcal{S}_{\mathrm{val}}$. Applying Hoeffding inequality together with the union bound over F

we conclude that with probability of at least $1 - \delta_2$,

$$\mathrm{risk}(f) \leq \mathrm{risk}(f^\star) + \sqrt{\frac{2\ln\left(m/\delta_2\right)}{m_{\mathrm{val}}}} \quad ,$$

where $m_{\mathrm{val}} = |\mathcal{S}_{\mathrm{val}}|$. We have therefore shown that with probability of at least $1 - \delta_1 - \delta_2$ the following inequality holds,

$$\mathrm{risk}(f) \;\leq\; \frac{1}{m}\sum_{i=1}^{m} \ell_i(\mathbf{w}^\star) + \frac{\|\mathbf{w}^\star\|^2}{m} + \frac{\sqrt{2\ln(1/\delta_1)}}{\sqrt{m}} + \frac{\sqrt{2\ln(m/\delta_2)}}{\sqrt{m_{\mathrm{val}}}} \quad .$$

Setting $\delta_1 = \delta_2 = \delta/2$ concludes our proof. $\square$

B Lists of Keywords

We give here the list of keywords used in the experiments described in Section 7.

The keywords used in the TIMIT experiments were: absolute, admitted, aligning, anxiety, apartments, apparently, argued, bedrooms, brand, camera, characters, cleaning, climates, controlled, creeping, crossings, crushed, decaying, demands, depicts, dominant, dressy, drunk, efficient, episode, everything, excellent, experience, family, firing, followed, forgiveness, freedom, fulfillment, functional, grazing, henceforth, ignored, illnesses, imitate, increasing, inevitable, introduced, January, materials, millionaires, mutineer, needed, obvious, package, paramagnetic, patiently, pleasant, possessed, pressure, radiation, recriminations, redecorating, rejected, secularist, shampooed, solid, spilled, spreader, story, strained, streamlined, street, stripped, stupid, superb, surface, swimming, sympathetically, unenthusiastic, unlined, urethane, usual, walking, weekday.

The keywords used in the HTIMIT experiments were: ambitious, appetite, avoided, bricks, building, causes, chroniclers, clinches, coeducational, colossal, concern, controlled, convincing, coyote, derived, desires, determination, disregarding, dwarf, effective, enrich, example, examples, excluded, executive, experiment, feverishly, firing, glossy, handle, happily, healthier, leaflet, lousiness, manure, misery, Nathan, northeast, notoriety, nutrients, obviously, overcame, penetrated, persuasively, petting, portion, precaution, prepare, prepared, privately, properties, propriety, reduced, referred, sandwich, sculptor, showering, sitting, sixty, sketched, skills, spirits, storm, strength, strip, surely, synagogue, technical, tomblike, traffic, tuna-fish, tycoons, university, vaguely, vanquished, virtues, waking, wedded, working, wounds.

The keywords used in WSJ experiment were: ability, administrative, analysis, answer, answer, business, business, children, children, clothes, company, confirmation, design, different, economy, economy, environment, environment, environment, equipment, evening, evening, experience, family, family, history, hospitalization, hospitalization, important, information, ingredients, interior, language, language, lesson, literature, literature, marketing, medicine, medicine, murder, murder, natural, necessary, newspaper, organizations, people, physical, physical, popular, popular, predispositions, preparation, private, procedure, process, progress, progress, psychological, public, public, questions, questions, reasons, regular, regular, research, research, responsibility, responsibility, scientists, sexual, simple, single, standards, strong, students, treatment, vegetable, violence.

The keywords used in OGI Stories experiment were: army, articles, available, baseball, boxing, bus, California, climbed, closed, competitive, contained, contributions, cost, course, crazy, creating, cutting, developed, directions, double, entertaining, experiencing, fee, fifty, forward, from, futures, Georgia, heading, innovation, institutional, interior, kick, kindness, land, listen, luck, Maine, main, much, never, nightly, nineteenth, operators, public, rancho, recession, recommendations, Robert, room, such, teach, technology, Texas, though, turning, understood, ways, western, yesterday.

References

Bahl, L., Brown, P., de Souza, P., Mercer, R., 1986. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In: Proc. of International Conference on Audio, Speech and Signal Processing. pp. 49–52.

Benayed, Y., Fohr, D., Haton, J.-P., Chollet, G., 2004. Confidence measure for keyword spotting using support vector machines. In: Proc. of International Conference on Audio, Speech and Signal Processing. pp. 588–591.

Bengio, S., Maréthoz, J., Keller, M., 2005. The expected performance curve. In: Proceedings of the 22nd International Conference on Machine Learning.

Bourlard, H., D'Hoore, B., Boite, J.-M., 1994. Optimizing recognition and rejection performance in wordspotting systems. In: Proc. of International Conference on Audio, Speech and Signal Processing. pp. 373–376.

Cardillo, P., Clements, M., Miller, M., 2002. Phonetic searching vs. LVCSR: How to find what you really want in audio archives. International Journal of Speech Technology 5, 9–22.

Cesa-Bianchi, N., Conconi, A., Gentile, C., September 2004. On the generalization ability of on-line learning algorithms. IEEE Transactions on Information Theory 50 (9), 2050–2057.

Collobert, R., Bengio, S., Mariéthoz, J., 2002. Torch: a modular machine learning software library. IDIAP-RR 46, IDIAP.

Cortes, C., Mohri, M., 2004. Confidence intervals for the area under the ROC curve. In: Advances in Neural Information Processing Systems 17.

Cortes, C., Vapnik, V., September 1995. Support-vector networks. Machine Learning 20 (3), 273–297.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y., 2006. Online passive aggressive algorithms. Journal of Machine Learning Research 7.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines. Cambridge University Press.

Dekel, O., Keshet, J., Singer, Y., 2004. Online algorithm for hierarchical phoneme classification. In: Workshop on Multimodal Interaction and Related Machine Learning Algorithms; Lecture Notes in Computer Science. Springer-Verlag, pp. 146–159.

Fu, Q., Juang, B.-H., 2007. Automatic speech recognition based on weighted minimum classification error (W-MCE) training method. In: IEEE Workshop on Automatic Speech Recognition & Understanding. pp. 278–283.

Juang, B.-H., Chou, W., Lee, C.-H., 1997. Minimum classification error rate methods for speech recognition. IEEE Transactions on Speech and Audio Processing 5 (3), 257–265.

Junkawitsch, J., Ruske, G., Höge, H., 1997. Efficient methods for detecting keywords in continuous speech. In: Proc. of European Conference on Speech Communication and Technology. pp. 259–262.

Keshet, J., Chazan, D., Bobrovsky, B.-Z., 2001. Plosive spotting with margin classifiers. In: Proceedings of the Seventh European Conference on Speech Communication and Technology. pp. 1637–1640.

Keshet, J., Shalev-Shwartz, S., Bengio, S., Singer, Y., Chazan, D., 2006. Discriminative kernel-based phoneme sequence recognition. In: Interspeech.

Keshet, J., Shalev-Shwartz, S., Singer, Y., Chazan, D., Nov. 2007. A large margin algorithm for speech and audio segmentation. IEEE Trans. on Audio, Speech and Language Processing.

Ketabdar, H., Vepa, J., Bengio, S., Bourlard, H., 2006. Posterior based keyword spotting with a priori thresholds. In: Prof. of Interspeech.

Manos, A., Zue, V., 1997. A segment-based wordspotter using phonetic filler models. In: Proc. of International Conference on Audio, Speech and Signal Processing. pp. 899–902.

Paul, D., Baker, J., 1992. The design for the Wall Street Journal-based CSR corpus. In: Proc. of the International Conference on Spoken Language Processing.

Platt, J. C., 1998. Fast training of Support Vector Machines using sequential minimal optimization. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods - Support Vector Learning. MIT Press.

Rabiner, L., Juang, B., 1993. Fundamentals of Speech Recognition. Prentice Hall.

Rahim, M., Lee, C., Juang, B., 1997. Discriminative utterance verification for connected digits recognition. IEEE Transactions on Speech and Audio Processing, 266–277.

Reynolds, D., 1997. HTIMIT and LLHDB: speech corpora for the study of handset transducer effects. In: Proc. of International Conference on Audio, Speech and Signal Processing. pp. 1535–1538.

Rohlicek, J. R., Jeanrenaud, P., Gish, K. N. H., Musicus, B., Siu, M., 1993. Phonetic training and language modeling for word spotting. In: Proc. of International Conference on Audio, Speech and Signal Processing. pp. 459–462.

Rohlicek, J. R., Russell, W., Roukod, S., Gish, H., 1989. Continuous hidden markov model for speaker independent word spotting. In: Proc. of International Conference on Audio, Speech and Signal Processing. pp. 627–430.

Rose, R., Paul, D., 1990. A hidden markow model based keyword recognition system. In: Proc. of International Conference on Audio, Speech and Signal Processing. pp. 129–132.

Salomon, J., King, S., Osborne, M., 2002. Framewise phone classification using support vector machines. In: Proceedings of the Seventh International Conference on Spoken Language Processing. pp. 2645–2648.

Shalev-Shwartz, S., Keshet, J., Singer, Y., 2004. Learning to align polyphonic music. In: Proceedings of the 5th International Conference on Music Information Retrieval.

Silaghi, M.-C., Bourlard, H., 1999. Iterative posterior-based keyword spotting without filler models. In: Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop. Keystone, USA, pp. 213–216.

Szoke, I., Schwarz, P., Matejka, P., Burget, L., Fapso, M., Karafiat, M., Cernocky, J., 2005. Comparison of keyword spotting approaches for informal continuous speech. In: Proc. of Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms.

Taskar, B., Guestrin, C., Koller, D., 2003. Max-margin markov networks. In: Advances in Neural Information Processing Systems 17.

Vapnik, V. N., 1998. Statistical Learning Theory. Wiley.

Weintraub, M., 1995. LVCSR log-likelihood ratio scoring for keyword spotting. In: Proc. of International Conference on Audio, Speech and Signal Processing. pp. 129–132.