# Phoneme Alignment Based on Discriminative Learning

*Joseph Keshet* — Hebrew University — jkeshet@cs.huji.ac.il

*Shai Shalev-Shwartz* — Hebrew University — shais@cs.huji.ac.il

*Yoram Singer* — Google Inc. — singer@google.com

*Dan Chazan* — IBM Haifa Labs — chazan@il.ibm.com

## Abstract

We propose a new paradigm for aligning a phoneme sequence of a speech utterance with its acoustical signal counterpart. In contrast to common HMM-based approaches, our method employs a discriminative learning procedure in which the learning phase is tightly coupled with the alignment task at hand. The alignment function we devise is based on mapping the input acoustic-symbolic representations of the speech utterance along with the target alignment into an abstract vector space. We suggest a specific mapping into the abstract vector-space which utilizes standard speech features (e.g. spectral distances) as well as confidence outputs of a framewise phoneme classifier. Building on techniques used for large margin methods for predicting whole sequences, our alignment function distills to a classifier in the abstract vector-space which separates correct alignments from incorrect ones. We describe a simple iterative algorithm for learning the alignment function and discuss its formal properties. Experiments with the TIMIT corpus show that our method outperforms the current state-of-the-art approaches.

## 1. Introduction

Phoneme alignment is the task of proper positioning of a sequence of phonemes in relation to a corresponding continuous speech signal. This problem is also referred to as phoneme segmentation. An accurate and fast alignment procedure is a necessary tool for developing speech recognition and text-to-speech systems.

Most previous work on phoneme alignment has focused on a generative model of the speech signal using Hidden Markov Models (HMM). See for example [1, 6, 14] and the references therein. Despite their popularity, HMM-based approaches have several drawbacks such as convergence of the EM procedure to local maxima and overfitting effects due to the large number of parameters. In this paper we propose an alternative approach for phoneme alignment that builds upon recent work on discriminative supervised learning. The advantage of discriminative learning algorithms 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 discriminative learning algorithms are likely to outperform generative models for the same task (cf. [15, 4]). One of the best known discriminative learning algorithms is the support vector machine (SVM), which has been successfully applied in speech applications [11, 7, 9]. The classical SVM algorithm is designed for simple decision tasks such as binary classification and regression. Hence, its exploitation in speech systems so far has also been restricted to simple decision tasks such as phoneme classification. The phoneme alignment problem is more involved, since we need to predict a sequence of phoneme start times rather than a single number. The main challenge of this paper is to extend the notion of discriminative learning to the complex task of phoneme alignment.

Our proposed method is based on recent advances in kernel machines and large margin classifiers for sequences [13, 12], which in turn build on the pioneering work of Vapnik and colleagues [15, 4]. The alignment function we devise is based on mapping the speech signal and its phoneme representation along with the target alignment into an abstract vector-space. Building on techniques used for learning SVMs, our alignment function distills to a classifier in this vector-space which is aimed at separating correct alignments from incorrect ones. We describe a simple iterative algorithm for learning the alignment function and discuss its formal properties. Experiments with the TIMIT corpus show that our method outperforms the best performing HMM-based approach [1].

This paper is organized as follows. In Sec. 2 we formally introduce the phoneme alignment problem. Our specific learning method is then described in Sec. 3. Next, we present experimental results in Sec. 4. Finally, concluding remarks and future directions are discussed in Sec. 5.

## 2. Problem Setting

In this section we formally describe the alignment problem. 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 $|\bar{\mathbf{x}}|$.

In the alignment problem, we are given a speech utterance along with a phonetic representation of the utterance. Our goal is to generate an alignment between the speech signal and the phonetic representation. The Mel-frequency cepstrum coefficients (MFCC) along with their first and second derivatives are extracted from the speech in the standard way which is based on the ETSI standard for distributed speech recognition. We denote the domain of the acoustic feature vectors by $\mathcal{X} \subset \mathbb{R}^d$. The acoustic feature representation of a speech signal is therefore a sequence of vectors $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathcal{X}$ for all $1 \leq t \leq T$. A phonetic representation of an utterance is defined as a string of phoneme symbols. Formally, we denote each phoneme by $p \in \mathcal{P}$, where $\mathcal{P}$ is the set of 48 English American phoneme symbols as proposed by [8]. Therefore, a phonetic representation of a speech utterance consists of a sequence of phoneme values $\bar{p} = (p_1, \ldots, p_k)$. Note that the number of phonemes clearly varies from one utterance to another and thus $k$ is not fixed. We denote by $\mathcal{P}^\star$ (and similarly $\mathcal{X}^\star$) the set of all finite-length sequences over $\mathcal{P}$. In summary, an alignment input is a pair $(\bar{\mathbf{x}}, \bar{p})$ where $\bar{\mathbf{x}}$ is an acoustic representation of the speech signal and $\bar{p}$ is a phonetic representation of the same signal. An alignment between the acoustic and phonetic representations of a spoken utterance is a sequence of start-times $\bar{y} = (y_1, \ldots, y_k)$ where $y_i \in \mathbb{N}$ is the start-time (measured as frame number) of phoneme $i$ in the acoustic sig-

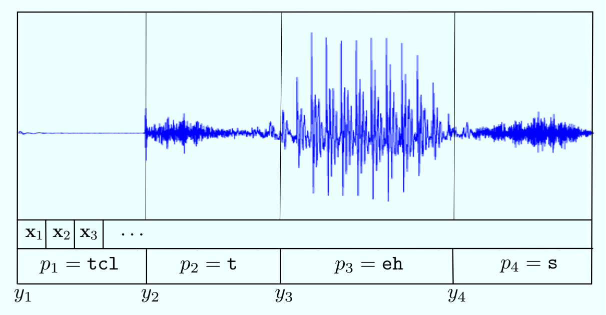

Figure 1: The first four phonemes of the word "test" taken from the TIMIT corpus. The speech signal is depicted in the upper part of the figure. The vertical lines indicate the TIMIT manual alignment.

nal. Each phoneme $i$ therefore starts at frame $y_i$ and ends at frame $y_{i+1} - 1$. An example of the notation described above is depicted in Fig. 1.

Clearly, there are different ways to pronounce the same utterance. Different speakers have different accents and tend to speak at different rates. Our goal is to learn an alignment function that predicts the true start-times of the phonemes from the speech signal and the phonetic representation.

To motivate our construction, let us take a short detour in order to discuss the common generative approaches for phoneme alignment. In the generative paradigm, we assume that the speech signal $\bar{\mathbf{x}}$ is generated from the phoneme sequence $\bar{p}$ and from the sequence of their start times $\bar{y}$, based on a probability function $\Pr[\bar{\mathbf{x}}|\bar{p},\bar{y}]$. The maximum posterior prediction is therefore,

$$\bar{y}' = \operatorname*{argmax}_{\bar{y}} \Pr[\bar{y}|\bar{\mathbf{x}},\bar{p}] = \operatorname*{argmax}_{\bar{y}} \Pr[\bar{y}|\bar{p}] \Pr[\bar{\mathbf{x}}|\bar{p},\bar{y}] \quad ,$$

where the last equality follows from Bayes rule. Put another way, the predicted $\bar{y}'$ is based on two probability functions:

I. a prior probability $\Pr[\bar{y}|\bar{p}]$.

II. a posterior probability $\Pr[\bar{\mathbf{x}}|\bar{p},\bar{y}]$.

To facilitate efficient calculation of $\bar{y}'$, practical generative models assume that the probability functions may be further decomposed into basic probability functions. For example, in the HMM framework it is commonly assumed that, $\Pr[\bar{\mathbf{x}}|\bar{p},\bar{y}] = \prod_i \prod_t \Pr[\mathbf{x}_t|p_i]$, and that, $\Pr[\bar{y}|\bar{p}] = \prod_i \Pr[\ell_i|\ell_{i-1}, p_i, p_{i-1}]$, where $\ell_i = y_{i+1} - y_i$ is the length of the $i$th phoneme according to $\bar{y}$.

These simplifying assumptions lead to a model which is quite inadequate for purpose of generating natural speech utterances. Yet, the probability of the sequence of phoneme start-times given the speech signal and the phoneme sequences is used as an assessment for the quality of the alignment sequence. The learning phase of the HMM aims at determining the basic probability functions from a training set of examples. The learning objective is to find functions $\Pr[\mathbf{x}_t|p_i]$ and $\Pr[\ell_i|\ell_{i-1}, p_i, p_{i-1}]$ such that the likelihood of the training set is maximized. Given these functions, the prediction $\bar{y}'$ is calculated in the so-called inference phase which can be performed efficiently using dynamic programming.

In this paper we describe and analyze an alternative paradigm in which the learning phase is tightly coupled with the decision task the algorithm must perform. Rather than working with probability functions we assume the existence of a predefined set of base alignment functions, $\{\phi_j\}_{j=1}^n$. Each base function takes the form $\phi_j : \mathcal{X}^\star \times (\mathcal{P} \times \mathbb{N})^\star \to \mathbb{R}$ . Thus, the input of each base function is an acoustic-phonetic representation, $(\bar{\mathbf{x}}, \bar{p})$, together with a candidate alignment $\bar{y}$. The base function returns a scalar which, intuitively, represents the confidence in the suggested alignment, $\bar{y}$. We denote by $\boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{y})$ the vector in $\mathbb{R}^n$ whose $j$th element is $\phi_j(\bar{\mathbf{x}}, \bar{p}, \bar{y})$. The alignment functions we use are of the form

$$f(\bar{\mathbf{x}}, \bar{p}) = \operatorname*{argmax}_{\bar{y}} \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{y}) \quad , \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^n$ is a vector of importance weights that must be learned. In words, $f$ returns a suggestion for an alignment sequence by maximizing a weighted sum of the scores returned by each base function $\phi_j$. Note that the number of possible alignment sequences is exponentially large. Nevertheless, as in the generative case, if the base functions $\phi_j$ are decomposable, the optimization in Eq. (1) can be efficiently calculated using a dynamic programming procedure.

As mentioned above, we would like to learn the function $f$ from examples. Each example is composed of an acoustic and a phonetic representation of an utterance $(\bar{\mathbf{x}}, \bar{p})$ together with the true alignment between them, $\bar{y}$. Let $\bar{y}' = f(\bar{\mathbf{x}}, \bar{p})$ be the alignment suggested by $f$. We denote by $\gamma(\bar{y}, \bar{y}')$ the cost of predicting the alignment $\bar{y}'$ where the true alignment is $\bar{y}$. Formally, $\gamma : (\mathbb{N} \times \mathbb{N})^\star \to \mathbb{R}$ is a function that gets two alignments and returns a scalar which is the cost of predicting $\bar{y}'$ where the true alignment is $\bar{y}$. We assume that $\gamma(\bar{y}, \bar{y}') \geq 0$ and that $\gamma(\bar{y}, \bar{y}) = 0$. An example for such a cost function is,

$$\gamma(\bar{y}, \bar{y}') = \frac{1}{|\bar{y}|} \sum_{i=1}^{|\bar{y}|} |y_i - y'_i| \quad .$$

The above cost is the average of the absolute differences between the predicted alignment and the true alignment. In our experiments, we used a variant of the above cost function and replaced the summands $|y_i - y'_i|$ with $\max\{0, |y_i - y'_i| - \varepsilon\}$, where $\varepsilon$ is a predefined small constant. The advantage of this cost is that no loss is incurred due to the $i$th phoneme if $y_i$ and $y'_i$ are within a distance $\varepsilon$ of each other. The goal of the learning process is to find an alignment function $f$ that attains small cost on unseen examples. In the next section we show how to use the training set in order to find an alignment function $f$ which with high probability, attains a small cost on the training set and on unseen examples as well.

## 3. The Learning Apparatus

In this section we present the details of our novel discriminative approach for phoneme alignment. Recall that our construction is based on a set of base alignment functions $\{\phi_j\}_{j=1}^n$ which maps an acoustic-phonetic representation of a speech utterance as well as a suggested alignment into an abstract vector-space. We start the section by introducing a specific set of base functions which is highly adequate for our phoneme alignment problem. Next, we describe a simple iterative procedure for finding a weight vector $\mathbf{w}$. The role of $\mathbf{w}$ is to rank the possible alignments for an input utterance such that the correct alignment attains the top rank.

### 3.1. Base Alignment Functions

We utilize seven different base functions ($n = 7$). These base functions are used for defining our alignment function $f(\bar{\mathbf{x}}, \bar{p})$ as in Eq. (1). To facilitate an efficient evaluation of $f(\bar{\mathbf{x}}, \bar{p})$ one must enforce structural constraints on the base functions. In the following, we describe our base functions while explicitly paying attention to their decomposability properties, which later enables us to efficiently evaluate $f(\bar{\mathbf{x}}, \bar{p})$ using a dynamic programming procedure. The same kind of structural assumptions are also assumed in HMM-based approaches.

Our first four base functions aim at capturing transitions between phonemes. These base functions are based on the distance between frames of the acoustical signal at two sides of phoneme boundaries as suggested by an alignment $\bar{y}$. The distance measure we employ, denoted $d$, 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 $4$ base functions are defined as,

$$\phi_s(\bar{\mathbf{x}}, \bar{p}, \bar{y}) = \sum_{i=1}^{|\bar{y}|} d(\mathbf{x}_{y_i - s}, \mathbf{x}_{y_i + s}), \quad s \in \{1, 2, 3, 4\} , \qquad (2)$$

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

The fifth base function we use is based on the frame-wise phoneme classifier described in [5]. Formally, for each phoneme $p \in \mathcal{P}$ 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 base function measures the cumulative confidence of the complete speech signal given the phoneme sequence and their start-times,

$$\phi_5(\bar{\mathbf{x}}, \bar{p}, \bar{y}) = \sum_{i=1}^{|\bar{p}|} \sum_{t=y_i}^{y_{i+1}-1} g_{p_i}(\mathbf{x}_t) \; . \qquad (3)$$

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

$$\phi_6(\bar{\mathbf{x}}, \bar{p}, \bar{y}) = \sum_{i=1}^{|\bar{y}|-1} \mathcal{N}(y_{i+1} - y_i; \hat{\mu}_{p_i}, \hat{\sigma}_{p_i}) \; , \qquad (4)$$

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 entire TIMIT training set, excluding SA1 and SA2 utterances.

Our last base function exploits assumptions on the speaking rate of a speaker. Intuitively, people usually speaks in an almost steady rate and therefore an alignment sequence in which speech rate is changed abruptly is probably incorrect. Formally, let $\hat{\mu}_p$ be the average length required to pronounce the $p$th phoneme. We denote by $r_i$ the relative speech rate, $r_i = (y_{i+1} - y_i)/\hat{\mu}_p$. That is, $r_i$ is the ratio between the actual length of phoneme $p_i$ as suggested by $\bar{y}$ 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 $(r_i - r_{i-1})^2$ and we define the base function $\phi_7$ as the cumulative sum of the changes in the speaking rate,

$$\phi_7(\bar{\mathbf{x}}, \bar{p}, \bar{y}) = \sum_{i=2}^{|\bar{y}|-1} (r_i - r_{i-1})^2 \; . \qquad (5)$$

We conclude the descriptions of the base alignment functions, with a discussion of the practical evaluation of the alignment function $f$. Recall that calculating $f$ requires solving the optimization problem, $f(\bar{\mathbf{x}}, \bar{p}) = \mathrm{argmax}_{\bar{y}} \, \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{y})$. A direct search for the maximizer is not feasible since the number of possible alignment sequences $\bar{y}$ is exponential in the length of the sequence. Fortunately, the base functions we have presented are decomposable and thus the best alignment sequence can be found in polynomial time using dynamic programming (similarly to the Viterbi procedure often implemented in HMMs [10]).

### 3.2. A Learning Algorithm

We now turn to the description of our iterative learning algorithm for phoneme alignment. Recall that a supervised learning algorithm for alignment receives as input a training set $S = \{(\bar{\mathbf{x}}_1, \bar{p}_1, \bar{y}_1), \ldots, (\bar{\mathbf{x}}_m, \bar{p}_m, \bar{y}_m)\}$ and returns a weight vector $\mathbf{w}$ defining the alignment function $f$ given by Eq. (1). Similar to the SVM algorithm for binary classification, our approach for choosing the weight vector $\mathbf{w}$ is based on the idea of large-margin separation. However, in our case, alignments are not merely correct or incorrect. Instead, the cost function $\gamma(\bar{y}, \bar{y}')$ is used for assessing the quality of alignments. Therefore, we do not aim at separating correct alignments from incorrect ones but rather try to rank alignments according to their quality. Theoretically, our approach can be described as a two-step procedure: First, we construct a vector $\phi(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}')$ in the vector space $\mathbb{R}^n$ based on each instance $(\bar{\mathbf{x}}_i, \bar{p}_i)$ in the training set $S$ and each possible alignment $\bar{y}'$. Second, we find a vector $\mathbf{w} \in \mathbb{R}^n$, such that the projection of vectors onto $\mathbf{w}$ ranks the vectors constructed in the first step above according to their quality. Formally, for each instance $(\bar{\mathbf{x}}_i, \bar{p}_i)$ and for each possible suggested alignment $\bar{y}'$, the following constraint should hold,

$$\mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}_i) - \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}') \;\geq\; \gamma(\bar{y}_i, \bar{y}') - \xi_i \; , \qquad (6)$$

where $\xi_i$ is a non-negative slack variable indicates the loss of the $i$th example. The SVM solution for the problem is therefore the weight vector $\mathbf{w} \in \mathbb{R}^n$ which minimizes the objective function $\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i$ while satisfying all the constraints in Eq. (6). The parameter $C$ serves as a complexity-accuracy trade-off parameter (see [4]).

In practice, the above two-step procedure can not be directly implemented since the number of constraints is exponentially large. To overcome this obstacle, we describe a simple iterative procedure for finding $\mathbf{w}$. Our iterative algorithm first constructs a sequence of weight vectors $\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_m$. The first weight vector is set to be the zero vector, $\mathbf{w}_0 = \mathbf{0}$. On iteration $i$ of the algorithm, we utilize the $i$th example of the training set along with the previous weight vector $\mathbf{w}_i$, for defining the next weight vector $\mathbf{w}_{i+1}$. Let $\bar{y}'$ be the predicted alignment sequence for the $i$th example according to $\mathbf{w}_i$. We set the next weight vector $\mathbf{w}_{i+1}$ to be the minimizer of the following optimization problem,

$$\begin{aligned} &\min_{\mathbf{w} \in \mathbb{R}^n, \xi \geq 0} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_i\|^2 + C\xi \qquad \text{s.t.} \\ &\mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{y}) - \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{y}') \geq \sqrt{\gamma(\bar{y}, \bar{y}')} - \xi \; . \end{aligned} \qquad (7)$$

|  | Training set size | Test set size | $t \leq 10$ ms | $t \leq 20$ ms | $t \leq 30$ ms | $t \leq 40$ ms |
|---|---|---|---|---|---|---|
| Discrim. Alignment | 650 or 3696 | 192 (core) | **79.7** | **92.1** | **96.2** | **98.1** |
| Brugnara *et al* [1] | 3696 | 192 (core) | 75.3 | 88.9 | 94.4 | 97.1 |
| Discrim. Alignment | 650 or 2336 | 1344 (entire) | **80.0** | **92.3** | **96.4** | **98.2** |
| Brugnara *et al* [1] | 2336 | 1344 (entire) | 74.6 | 88.8 | 94.1 | 96.8 |

Table 1: Percentage of correctly positioned boundaries, given a predefined tolerance

This optimization problem can be thought of as a relaxed version of the SVM optimization problem with three major differences. First, we replace the exponential number of constraints from Eq. (6) with a single constraint. This constraint is based on the predicted alignment $\bar{y}'$ according to the previous weight vector $\mathbf{w}_i$. Second, we replaced the term $\|\mathbf{w}\|^2$ in the objective function of the SVM with the term $\|\mathbf{w} - \mathbf{w}_i\|^2$. Intuitively, we would like to minimize the loss of $\mathbf{w}$ on the current example, i.e., the slack variable $\xi$, while remaining as close as possible to our previous weight vector $\mathbf{w}_i$. Last, we replace $\gamma(\bar{y}, \bar{y}')$ with $\sqrt{\gamma(\bar{y}, \bar{y}')}$ for technical reasons which will be given elsewhere. It can be shown (see [3]) that the solution to the above optimization problem is, $\mathbf{w}_{i+1} = \mathbf{w}_i + \min\{\ell/\|\mathbf{a}\|^2, C\}\,\mathbf{a}$, where $\mathbf{a} = \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}_i) - \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}')$ and $\ell = \max\{(\gamma(\bar{y}_i, \bar{y}'))^{1/2} - \mathbf{w}_i \cdot \mathbf{a}, 0\}$.

The above iterative procedure gives us a sequence of weight vectors. We briefly note that it has been proved that at least one of the resulting alignment functions is likely to have good generalization properties [12, 2]. To find an alignment function that generalizes well, we calculate the average cost of each alignment function on a validation set and choose the one that achieves the best results.

## 4. Experimental Results

To validate the effectiveness of the proposed approach we performed experiments with the TIMIT corpus. We first divided the training portion of the TIMIT (excluding the SA1 and SA2 utterances) into three disjoint parts containing 500, 100 and 3093 utterances. The first part of the training set was used for learning the functions $g_p$ (Eq. (3)), which define the base function $\phi_5$. Those functions were learned by the algorithm described in [5] using the MFCC+$\Delta$+$\Delta\Delta$ acoustic features and a Gaussian kernel ($\sigma = 6.24$ and $C = 5.0$). The second set of 100 utterances formed the validation set needed for our alignment algorithm as described in Sec. 3. Finally, we ran our iterative alignment algorithm on the remaining utterances in the training set. The value of $\varepsilon$ in the definition of $\gamma$ was set to be 1 (i.e., 10 ms).

We evaluated the learned alignment functions on both the core test set and the entire test set of TIMIT. A comparison of our results with the results reported in [1] is provided in Tab. 3.1. For each tolerance value $\tau \in \{10\,\text{ms}, 20\,\text{ms}, 30\,\text{ms}, 40\,\text{ms}\}$, we counted the number of predictions whose distance to the true boundary, $t = |y_i - y_i'|$, is less than $\tau$. As can be seen, our discriminative method outperforms the generative approach described in [1] on all predefined tolerance values. Furthermore, the results obtained by our algorithm are the same whether we use the entire 3093 utterances or only the first 50 utterances.

## 5. Discussion

We describe and experimented with a discriminative method for phoneme alignment. The proposed approach is based on recent advances in large margin classifiers. Our training algorithm is simple to implement and entertains convergence guarantees. In contrast to HMM training procedures which are prone to local maxima variabilities, our proposed algorithm is guaranteed to converge to a solution which has good generalization properties under mild conditions. Indeed, the experiments reported above suggest that the discriminative training requires fewer training examples than an HMM-based alignment procedure while achieving the best reported results for this task.

**Acknowledgements** This work was supported by EU project PASCAL Network Of Excellence No. IST-2002-506778.

## 6. References

[1] F. Brugnara, D. Falavigna, and M. Omologo. Automatic segmentation and labeling of speech based on hidden Markov models. *Speech Comm.*, 12:357–370, 1993.

[2] N. Cesa-Bianchi, A. Conconi, and C.Gentile. On the generalization ability of on-line learning algorithms. In *NIPS*, 2002.

[3] K. Crammer, O. Dekel, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. In *NIPS*, 2003.

[4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge Univ. Press, 2000.

[5] O. Dekel, J. Keshet, and Y. Singer. Online algorithm for hierarchical phoneme classification. In *MLMI*, 2004.

[6] J.-P. Hosom. Automatic phoneme alignment based on acoustic-phonetic modeling. In *ICSLP*, 2002.

[7] J. Keshet, D. Chazan, and B.-Z. Bobrovsky. Plosive spotting with margin classifiers. In *EUROSPEECH*, 2001.

[8] K.-F. Lee and H.-W. Hon. Speaker independent phone recognition using hidden Markov models. *IEEE Trans. Acous., Speech. and Signal Processing*, 37(2):1641–1648, 1989.

[9] Z. Litichever and D. Chazan. Classification of transition sounds with application to automatic speech recognition. In *EUROSPEECH*, 2001.

[10] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[11] J. Salomon, S. King, and M. Osborne. Framewise phone classification using support vector machine. *ICSLP*, 2002.

[12] S. Shalev-Shwartz, J. Keshet, and Y. Singer. Learning to align polyphonic music. In *ISMIR*, 2004.

[13] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS 17*, 2003.

[14] D.T. Toledano, L.A.H. Gomez, and L.V. Grande. Automatic phoneme segmentation. *IEEE Trans. Speech and Audio Proc.*, 11(6):617–625, 2003.

[15] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.