FIXED-DIMENSIONAL ACOUSTIC EMBEDDINGS OF VARIABLE-LENGTH SEGMENTS IN LOW-RESOURCE SETTINGS

Keith Levin,[1] *Katharine Henry,*[2] *Aren Jansen,*[1] *Karen Livescu*[3]

[1] Human Language Technology Center of Excellence & The Center for Language and Speech Processing
Johns Hopkins University, Baltimore, MD 21218 USA
[2]University of Chicago, Chicago, IL 60637 USA
[3]Toyota Technological Institute at Chicago, Chicago, IL 60637 USA

ABSTRACT

Measures of acoustic similarity between words or other units are critical for segmental exemplar-based acoustic models, spoken term discovery, and query-by-example search. Dynamic time warping (DTW) alignment cost has been the most commonly used measure, but it has well-known inadequacies. Some recently proposed alternatives require large amounts of training data. In the interest of finding more efficient, accurate, and low-resource alternatives, we consider the problem of embedding speech segments of arbitrary length into fixed-dimensional spaces in which simple distances (such as cosine or Euclidean) serve as a proxy for linguistically meaningful (phonetic, lexical, etc.) dissimilarities. Such embeddings would enable efficient audio indexing and permit application of standard distance learning techniques to segmental acoustic modeling. In this paper, we explore several supervised and unsupervised approaches to this problem and evaluate them on an acoustic word discrimination task. We identify several embedding algorithms that match or improve upon the DTW baseline in low-resource settings.

Index Terms— Fixed-dimensional embedding, segmental acoustic modeling, query-by-example search, speech indexing

1. INTRODUCTION

Most approaches to speech recognition and related tasks to date have handled variability in word and phone segment duration by modeling short, fixed-length frames. These approaches rely on frame-level independence assumptions whose limited validity has been well documented [1]. To account for more acoustic context, many recognition systems, including recent sparse exemplar models [2], compute supervectors of concatenated acoustic features over longer (but still fixed) windows, often followed by dimensionality reduction. Still, variation in segment duration prevents these fixed-context windows from always aligning with meaningful linguistic units.

In contrast, template-based and segmental approaches use variable-length acoustic windows to capture whole units for subsequent modeling. Template-based acoustic models typically rely on dynamic time warping (DTW) to quantify the similarity of phone or word segments [3, 4]. However, DTW often misestimates word segment similarity due to, among other factors, oversensitivity to longer phonetic segments (e.g. vowels). Furthermore, DTW alignment is polynomial time in the duration of the segments being compared, which can prove burdensome when comparing test audio to a large repository of exemplars. This drawback could be avoided by embedding arbitrary-length segments into fixed-dimensional spaces in which common distances provide estimates of linguistic dissimilarity. Such embeddings would (i) enable the application of standard distance learning techniques [5, 6] to template-based acoustic modeling and (ii) support a new generation of efficient segment-based audio indexing algorithms, enabling highly scalable spoken term discovery [7, 8, 9] and query-by-example search [10, 11, 12].

Existing segmental acoustic models use fixed-dimensional representations of hypothesized variable-length segments. The various flavors of segmental models provide several ways of constructing these representations. These include downsampling [13, 14, 15, 16], phonetic acoustic model-derived features [17, 18], and convolutional deep neural networks [19]. These techniques do not necessarily produce linguistically meaningful embeddings but rather rely on supervision in the segmental feature space for linguistic discrimination. Furthermore, with the exception of basic downsampling, these approaches do not extend well to zero- or low-resource settings, where supervised training data is limited or non-existent.

With these motivations, we explore multiple unsupervised and supervised approaches to extracting fixed-dimensional embeddings of variable-length audio signals, focusing for the time being on signals corresponding to individual words. Our goal is to identify embeddings that preserve word discrimination under simple cosine or Euclidean distances. To apply our techniques to large amounts of speech, we require built-in out-of-sample extension capabilities. We consider three operational settings in which we have access to varying levels of information. At one extreme, we assume that we see each unlabeled speech segment in isolation with no additional training data. Here we can consider downsampling methods. At the opposite extreme, with a training set of word exemplars of known type we can learn feature maps that maintain word type discrimination. Finally, in the intermediate case, we have a training set of segments of unknown types, but we can still exploit the class-independent distribution of the exemplars. In each case, we explore both linear and non-linear embeddings and evaluate their effectiveness on a word type discrimination task in a multi-speaker corpus of conversational telephone speech. In all cases, we consider only low-resource settings (no more than several hours of speech).

2. METHODS

Our goal is to define a function that maps audio signals of arbitrary length to a continuous vector space that parsimoniously encodes the underlying linguistic content. Formally, let $\mathcal{X}$ denote the set of all arbitrary-length acoustic vector time series, $\mathcal{X} = \{X = x_1 x_2 \dots x_T \mid T \in \mathbb{Z}^+, \}$, with each $x_t \in \mathbb{R}^p$, where p is the dimensionality of some frame-level acoustic feature representation (e.g. MFCC, PLP). We would like to learn functions $f : \mathcal{X} \to \mathbb{R}^d$ that

map acoustic feature vector time series in $\mathcal{X}$ to points in $\mathbb{R}^d$ such that $f(X)$ and $f(Y)$ are similar if and only if X and Y are acoustic observations generated by similar linguistic units (e.g., phones, morphemes, syllables, words). For now we restrict the discussion and experiments to word segments, but the methods apply similarly to any meaningful unit. We consider three settings for learning these functions, relying on varying amounts of available information:

1. (*NoTrain*) We may access each test word segment $X \in \mathcal{X}$ in isolation with no additional information.
2. (*UnsupTrain*) We have a collection of N_{train} word exemplars $\mathcal{X}_{\text{train}} = \{X_i\}_{i=1}^{N_{\text{train}}}$, with each $X_i \in \mathcal{X}$.
3. (*SupTrain*) In addition to a collection of N_{train} word segments $\mathcal{X}_{\text{train}} \subset \mathcal{X}$, we have the corresponding word labels $\mathcal{W}_{\text{train}} = \{w_i\}_{i=1}^{N_{\text{train}}}$ for those word segments.

In what follows, we define approaches for these three settings.

2.1. Time series downsampling

If no information is available to us aside from a given feature vector time series, we must adopt strategies to select a fixed-sized set of observations. The simplest approach is to uniformly downsample so that any segment is represented by a constant number k of vectors: given a feature vector time series $X = x_1 x_2 \dots x_T \in \mathcal{X}$, with each $x_t \in \mathbb{R}^p$, we sample vectors from X at intervals of T/k with suitable interpolation as needed. The downsampled time series is concatenated into a single vector of dimensionality $d = kp$. A more sophisticated approach is to perform non-uniform downsampling of the time series using HMMs. For a segment $X = x_1 x_2 \dots x_T \in \mathcal{X}$, we train a k-state left-to-right HMM, modeling the acoustics with a single spherical Gaussian in each state. This approach segments X non-uniformly into k regions. Concatenating the means of these regions into a single vector yields an embedding into $\mathbb{R}^{kp}$ regardless of the length of X. While we restrict our experiments to this HMM-based approach, other HMM-based techniques may be applicable to our setting (e.g. see [20]), as may other non-uniform downsampling approaches (e.g. see [13, 14]).

2.2. Vector of distances to reference set

When we have access to a collection of training word exemplars $\mathcal{X}_{\text{train}}$, we can consider more sophisticated embedding techniques. Here, we designate a reference set of r exemplars, $\mathcal{X}_{\text{ref}} = \{X_{m_i} | 1 \leq m_i \leq N_{\text{train}}, i = 1, \dots, r\} \subseteq \mathcal{X}_{\text{train}}$, that covers a broad selection of word types and speakers. Given a feature vector time series $X \in \mathcal{X}$, we form a vector $u \in \mathbb{R}^r$ with the i^{th} component of u given by $D(X, X_{m_i})$, where $D(\cdot, \cdot)$ is the DTW alignment cost between pairs of segments. We refer to u as a *reference vector* for segment X. Note that this is a special case of a Lipschitz embedding in which each reference set has cardinality one [21] and that we use the term *reference set* in a slightly different sense. We can think of this reference vector as representing a word in terms of its similarity to a set of exemplars that forms a "basis" for the space of all words. Thus, this and similar such representations can be applied even to word types not seen in the training set.

One of our motivations for deriving fixed-dimensional word embeddings is to avoid costly DTW alignments over large collections of speech. Here, we are explicitly constructing a representation that requires computing DTW alignment cost against a set of reference examples. While this is an expensive operation, it is still linear in the size of the speech collection if the reference set is fixed. In the context of indexing for search applications, these DTW calculations need only be performed once offline for the entire search collection, allowing sublinear-time search using approximate nearest neighbor techniques [22]. As commonly employed for costly Lipschitz embeddings, inducing sparsity would also mitigate the computational burden (e.g. see [23]). In general, the approaches presented here replace DTW alignments with simple Euclidean or cosine distance computations. Thus, letting m and n be the lengths of the vector time series being aligned and letting p be the dimensionality of the vectors in the time series, we replace an operation requiring time $O(mnp)$ with an operation requiring time $O(d)$, where d is the dimensionality of our embedding. Thus, when using the techniques in [22] to search for a query term of length m in a vector time series of length N, we require only $O(\log N)$ time using approximate nearest neighbor search, rather than $O(Nmp)$ operations required by DTW-based search.

2.3. Linear embedding techniques

Linear dimensionality reduction techniques use a collection of labeled or unlabeled data to derive a linear map from the original feature space to a space of lower dimensionality. Applying such techniques to the reference vectors defined in Section 2.2, we obtain a projection matrix $P \in \mathbb{R}^{d \times r}$, where $d < r$. Given a new segment $X \in \mathcal{X}$, we project its reference vector $u \in \mathbb{R}^r$ to $u' = Pu \in \mathbb{R}^d$. In the absence of word type information, we may derive P using principal components analysis (PCA). If word labels are available, supervised techniques such as linear discriminant analysis (LDA) can be used. Note that if we use Euclidean distance to compare embedded segment pairs, then operating in the linear embedding space defined by projection matrix P is equivalent to using a Mahalanobis distance parametrized by matrix $M = P^T P$ in the original r-dimensional space.

2.3.1. PCA and LDA

PCA is a well-established unsupervised dimensionality reduction technique. Given $\mathcal{X}_{\text{train}} \subset \mathcal{X}$, we construct the reference vector of each $X_i \in \mathcal{X}_{\text{train}}$. The $d < r$ top (largest-magnitude eigenvalue) eigenvectors of the resulting covariance matrix form a basis of lower dimensionality that best preserves the variance of the data.

When we have word type labels $\mathcal{W}_{\text{train}} = \{w_1, \dots, w_{N_{\text{train}}}\}$ for the training exemplars, multi-class LDA can be used. Multi-class LDA finds a set of vectors pointing along the directions in which between-class variability is maximized while within-class variability is minimized. Specifically, we form a basis of the first d largest-eigenvalue non-trivial solutions v to the generalized eigenproblem $\Sigma_{\text{B}} v = \lambda \Sigma_{\text{W}} v$, where Σ_{B} and Σ_{W} are the between- and within-class covariance matrices of the training data, respectively. In our implementation, we regularize the within-class covariance matrix with shrinkage by adding a scaled identity matrix.

2.3.2. Metric learning to rank (MLR)

Another supervised option is to use one of many existing techniques for discriminatively learning a Mahalanobis distance, given by a positive semidefinite matrix M, with distance between vectors u_1, u_2 defined as $\sqrt{(u_1 - u_2)^T M (u_1 - u_2)}$. Here we use MLR [24], as it optimizes a criterion closely related to our task. MLR is a large-margin approach that aims to separate vectors that are similar to a given query vector from those that are dissimilar by a margin given by a ranking loss, which in our case is mean average precision Given the learned matrix M, we find a matrix U whose i^{th} row is $\sqrt{|\lambda_i|} v_i$,

where v_i is the i^{th} eigenvector of M with corresponding eigenvalue λ_i. We obtain projection matrix P by retaining only the first d rows of U.

2.4. Nonlinear graph embedding techniques

Numerous non-linear dimensionality reduction techniques are available for consideration (e.g. [25, 26]). We use Laplacian eigenmaps [27], including a variant proposed in [28] that defines an out-of-sample extension. In the supervised setting, we can encode word type information by adding graph edges that reflect word identity.

2.4.1. Laplacian eigenmaps

We begin by constructing a graph G with one vertex per training example and edges reflecting the nearest neighbor structure under DTW alignment cost. The binary-valued adjacency matrix $A^{\text{nn}} \in \mathbb{R}^{N_{\text{train}} \times N_{\text{train}}}$ has $A^{\text{nn}}_{ij} = 1$ if and only if example i is one of the k nearest neighbors of example j or vice versa. Given matrix A^{nn}, the *normalized* graph Laplacian operator is defined as $L^{\text{nn}} = I - S^{\frac{1}{2}} A^{\text{nn}} S^{\frac{1}{2}}$, where S is diagonal with $S_{ii} = \sum_j A^{\text{nn}}_{ij}$. Following [27], we wish to find a set of d projection maps $\{h_1, \ldots, h_d\}$, where $h_i : V(G) \rightarrow \mathbb{R}$, such that vertices near one another under the topology of G are mapped to similar locations in $\mathbb{R}^d$. Since the graph Laplacian operator acts as a measure of smoothness of functions defined on the graph, the desired set $\{h_i\}$ is defined implicitly by the eigenvectors of L^{nn} with the d smallest eigenvalues (after discarding the first trivial eigenvector, which has eigenvalue 0). Each eigenvector encodes the image of the vertex set under a map in $\{h_i\}$.

A problem arises when we wish to project a segment with no corresponding vertex in G into this d-dimensional space. Like multidimensional scaling, without some procedure for out-of-sample extension, this technique has little practical utility. An out-of-sample solution for Laplacian eigenmaps is given in [28] and is summarized below. We construct matrices A^{nn} and L^{nn} as described above. Our new optimization problem takes the form

$$h^* = \arg \min_{h \in \mathcal{H}_K} \mathbf{h}^T L^{\text{nn}} \mathbf{h} + \xi \|h\|^2_K, \tag{1}$$

where $\mathcal{H}_K$ is the reproducing kernel Hilbert space for some positive semi-definite kernel function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, $\mathbf{h} = \langle h(X_1), \ldots, h(X_{N_{\text{train}}}) \rangle^T$ is the vector of values of h computed on the vertices of the graph, and ξ is a non-negative regularization term. All reported experiments used a kernel function of the form

$$K(X_i, X_j) = \exp\left\{ -\frac{[\max(0, D(X_i, X_j) - \eta)]^2}{2\sigma^2} \right\},$$

where $D(\cdot, \cdot)$ is DTW alignment cost and $\eta, \sigma \in \mathbb{R}$. By the RKHS representer theorem [28], the j^{th} component of our projection map is

$$h^*_j(X) = \sum_{i=1}^{N_{\text{train}}} \alpha^{(j)}_i K(X_i, X), \tag{2}$$

where the $\{\alpha^{(j)}_i\}$ are given by solutions to the generalized eigenvector problem $(L^{\text{nn}} K + \xi I)\alpha = \lambda K \alpha$, with K being the Gram matrix with entries $K_{ij} = K(X_i, X_j)$ for $X_i, X_j \in \mathcal{X}_{\text{train}}$.

Intuitively, this eigenproblem is trying to find mappings from $\mathcal{X}_{\text{train}}$ to $\mathbb{R}$ such that word exemplars that are connected in graph G take similar values. In the out-of-sample extension, the kernelization performs a sort of interpolation such that a test exemplar "similar" to a vertex in G takes a similar value. Given the d eigenvectors with the smallest eigenvalues (ignoring the trivial one, as above), we can map an arbitrary segment $X \in \mathcal{X}$ to a point $v \in \mathbb{R}^d$ given by $v = \langle h_1(X), \ldots, h_d(X) \rangle^T$ according to Equation 2.

2.4.2. Supervised graph embedding

When available, it is desirable to incorporate class label information into the Laplacian eigenmaps approach. Notable recent algorithms for this problem include locality preserving discriminant analysis [29], locality sensitive discriminant analysis (LSDA) [30], and marginal Fisher analysis [31]. In our approach, we construct kernel matrix K and matrix A^{nn} as described above. Additionally, we construct a matrix A^{sup} such that $A^{\text{sup}}_{ij} = 1$ if $i \neq j$ and $w_i = w_j$, and $A^{\text{sup}}_{ij} = 0$ if $w_i \neq w_j$ or if $i = j$. Thus, A^{sup} captures our knowledge of which pairs of words ought to be adjacent to one another in an "ideal" graph reflecting the true class labels. We can combine our supervised and unsupervised information into a single graph Laplacian $L = L^{\text{nn}} + \beta L^{\text{sup}}$, $\beta \in \mathbb{R}$ is non-negative and L^{nn} and L^{sup} are the normalized graph Laplacians of A^{nn} and A^{sup}, respectively. L captures both acoustic similarity and true word label information in a single operator. This is analogous to LSDA, but where we linearly combine the normalized Laplacians of within- and between-class graphs rather than the adjacency matrices. Replacing L^{nn} with L, we proceed as in the previous algorithm, constructing a subspace from the first d non-trivial solutions to Equation 1.

2.4.3. LDA applied to graph embeddings

We again assume that we have a labeled set of vector time series, which we use to learn an embedding into $\mathbb{R}^{d'}$ using Laplacian eigenmaps as described above. This map is applied to the training set exemplars and an LDA projection is learned from the resulting vectors and their labels to produce a final embedding into $\mathbb{R}^d$. This two-step process provides an alternate means of introducing supervision into the graph embedding framework. We note that other supervised projections could also be used here, e.g. via Mahalanobis distance learning as in Section 2.3.2, but here we limit ourselves to LDA.

3. EXPERIMENTS

To evaluate the techniques described above, we use the task in [32], designed to evaluate the word discrimination performance of acoustic front ends and acoustic models that do not explicitly model phones. An evaluation set of presegmented words $\mathcal{X}_{\text{test}}$ is presented. For each pair $(X_i, X_j) \in \mathcal{X}_{\text{test}} \times \mathcal{X}_{\text{test}}$ for $i \neq j$, we compute $D(X_i, X_j)$ under the representation and distance D being evaluated. We set a threshold τ such that we declare words X_i and X_j to be the same if $D(X_i, X_j) \leq \tau$ and declare them to be different otherwise. Discriminative power is then quantified by the average precision (AP), the area under the precision-recall curve, which characterizes discrimination performance at all possible settings of τ. Let $N_{\text{SW}}(\tau)$ denote the number of same-label word pairs with distance less than or equal to τ under the model. We define the model's precision $P_{\text{SW}}(\tau)$ and recall $R_{\text{SW}}(\tau)$ at operating threshold τ as

$$P_{\text{SW}}(\tau) = \frac{N_{\text{SW}}(\tau)}{N(\tau)} \qquad R_{\text{SW}}(\tau) = \frac{N_{\text{SW}}(\tau)}{N_{\text{SW}}}, \tag{3}$$

where $N(\tau)$ denotes the total number of word pairs in the corpus whose distance under the model is less than or equal to τ (i.e., the number of hypothesized same-word pairs) and N_{SW} is the number of true same-word pairs in the corpus. Thus, to evaluate one of our candidate algorithms, we embed the test set according to that algorithm, compute all pairwise distances between the embedded points and compute the area under the precision-recall curve.

We assembled two collections of words from the Switchboard English corpus, $\mathcal{X}_{\text{train}}$ and $\mathcal{X}_{\text{test}}$, containing $N_{\text{train}} = 10383$ and $N_{\text{test}} = 11024$ words, respectively. Both sets were constrained to

Table 1. Average precision scores achieved by our baseline algorithms in the *NoTrain* condition, by feature type (all scores are given as proportions).

		Ave. Prec.	
Algorithm		**PLP**	**FDLP**
Baseline DTW		0.198	0.226
Uniform Downsampling	$n = 5$	0.036	0.040
	$n = 10$	0.062	0.069
	$n = 25$	0.072	0.081
	$n = 50$	0.074	0.082
Non-uniform Downsampling	$n = 5$	0.050	0.033
	$n = 10$	0.086	0.080
	$n = 25$	0.081	0.088
	$n = 50$	0.076	0.086

include only words of 6 or more orthographic characters and to be at least 50 frames in length (0.5 s). The train and test sets contained 5539 and 3392 word types, respectively, with 6971 unique word types in all. The train set was constructed to have a broad sampling of word types, with at most 5 tokens of any given word type and with each token of a given type taken from a different speaker. The resulting word set covered 360 conversation sides and 156 unique speakers. The test set was identical to that in [32]. It was constructed to reflect a content word distribution encountered in a typical conversational speech setting. It consisted of all words meeting the above length criteria from 360 conversation sides covering 236 unique speakers, none of whom appeared in the train set. To investigate the effect of acoustic front end on this task, we performed this evaluation using vector time series of 39-dimensional perceptual linear prediction (PLP) feature vectors and 15-dimensional truncated frequency-domain linear prediction (FDLP) feature vectors [33]. Previous work has indicated that truncating the spectrum from 13 to 5 dimensions yields a gain in this task relative to front ends with more detailed spectral content [34]. Cosine distance, defined for vectors a, b as $1 - a^T b / \|a\| \|b\|$, generally outperformed Euclidean distance for the embedding techniques described in this paper. The basic reference vector and PCA experiments used Euclidean distance between embedded points. All other experiments used cosine distance.

3.1. Baselines (the *NoTrain* condition)

Using DTW alignment cost as an interword distance measure establishes a baseline for our task. A successful algorithm will be one that can improve upon this result or maintain comparable performance without supervision while being computationally less expensive. Table 1 shows the performance of this baseline approach on both PLP and FDLP acoustic features. Also listed in Table 1 are the results using uniform and nonuniform downsampling approaches outlined in 2.1, where we consider target sample sizes of $n \in \{5, 10, 25, 50\}$ and use cosine distance to compare the resulting supervectors. As is the case for the DTW baseline, the downsampling results using FDLP are consistently comparable to or better than PLP. The gains of nonuniform sampling over uniform are marginal, with the best downsampling APs roughly 1/3 that of the baseline DTW performance for $n \geq 10$.

3.2. Unsupervised embeddings (the *UnsupTrain* condition)

Next we evaluated the reference vectors described in Section 2.2. A drawback of this approach (and the approaches that depend on it) is that constructing an acoustic segment's reference vector requires computing $|\mathcal{X}_{\text{ref}}| = r$ DTW alignment costs. Lower-dimensional reference vectors, if still effective in distinguishing words, would allow us to maintain similar performance with fewer DTW calculations required to embed a given word. To examine this possibility, we selected reference sets $\mathcal{X}_{\text{ref}} \subseteq \mathcal{X}_{\text{train}}$ of various sizes r. Reference sets were selected randomly, but biased to favor selecting clusters of same-word tokens. As reflected in Table 2, these results fall short of the baseline DTW scores, but they do demonstrate that we can safely shrink the size of our reference set by as much as a factor of 20 without paying too large a penalty in performance. We leave the problem of optimal reference set design for future work.

Table 2. Average precision scores achieved by our basic reference vectors in the *UnsupTrain* condition, by feature type (all scores are proportions).

	Ave. Prec.	
r	**PLP**	**FDLP**
100	0.041	0.078
500	0.089	0.137
1,000	0.089	0.142
5,000	0.094	0.149
10,000	0.096	0.150

We constructed train set reference vectors using a reference set of size r =10,000. We applied PCA to these reference vectors, and applied the learned projection to the test set reference vectors for evaluation. To apply Laplacian eigenmaps to our data, we first calculated all pairwise DTW alignment costs for words in $\mathcal{X}_{\text{train}}$ and, based on those costs, assembled the adjacency matrix A^{nn} and Gram matrix K as described in Section 2. Laplacian eigenmaps require setting certain parameters in addition to the target space dimensionality. Performance was reasonably stable for the number of nearest neighbors (k), the regularizer weight (ξ), and the kernel function parameters (η,σ) in the ranges $k \in [7, 30]$, $\xi \in [0.001, 0.1]$, $\eta \in [0.01, 0.05]$, and $\sigma \in [0.15, 0.04]$. We report results for the best-performing parameter settings, leaving the challenge of automatic selection for future work.

Figure 1(a) shows the performance of the unsupervised techniques outlined in Section 2 for varying target space dimensionalities. We find that using PCA, we can reduce dimension from 10,000 to 100 without substantial loss in performance, but overall performance falls short of the DTW baseline. Laplacian eigenmaps matches the DTW baseline for target dimensionalities $d > 100$ and greatly surpasses PCA at all target dimensionalities, indicating a more efficient use of dimensions than is possible with unsupervised linear methods.

3.3. Supervised embeddings (the *SupTrain* condition)

Analogously to PCA, multi-class LDA and MLR were performed on the train set reference vectors with word types as class labels. [1] The resulting linear projections were applied to the test set reference vectors for evaluation. We used a reference set of size $r = 10,000$, except for MLR applied to FDLP features, where we used $r = 5,000$. LDA performance depended moderately on the shrinkage scale factor, observing a change of up to 0.1 AP as we varied the scale factor from 0 to 5. All reported results used a scale factor of 1. MLR results depended moderately on the slack parameter, with typical good values in the range $[10^3, 10^5]$. Supervised graph-based embeddings were obtained using the procedure described in Section 2.4.2. Using the optimal parameter settings for Laplacian eigenmaps and varying β, we found that performance was stable for $\beta \geq 1$, indicating

[1] We used Brian McFee's implementation of MLR, available at `https://github.com/bmcfee/mlr/`

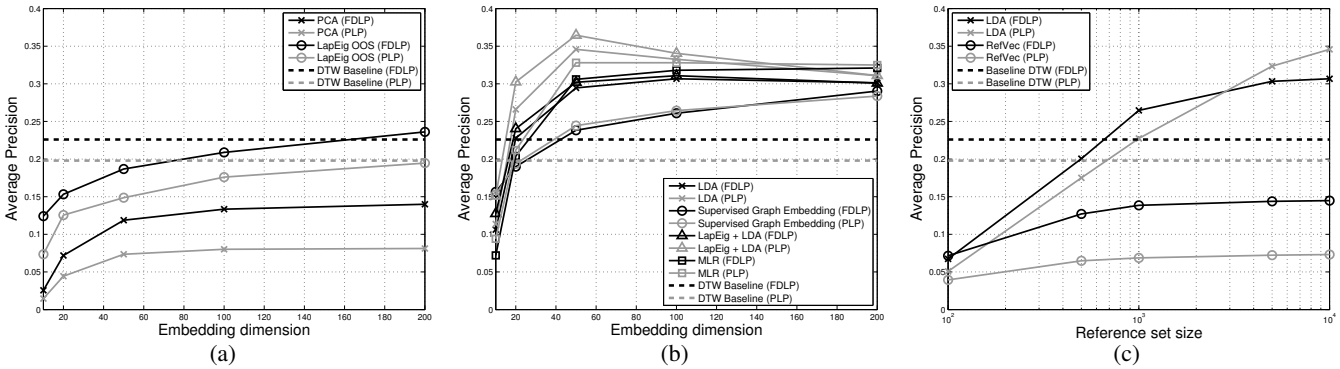

Fig. 1. Average precision as a function of target space dimensionality for (a) unsupervised embeddings (*UnsupTrain*) and (b) supervised embeddings (*SupTrain*), and (c) as a function of reference set size.

that the utility of supervision dominates that of the nearest neighbor graph structure. Finally, LDA was also applied to the Laplacian eigenmaps embeddings, with the projection again learned on the training set and evaluated on the test set.

Figure 1(b) shows the performance of the supervised techniques from Section 2 for varying target space dimensionalities. We find that LDA and MLR greatly improve upon the DTW baselines, with AP stable down to 50 dimensions. Interestingly, with supervision the 39-dimensional PLP features usually outperform the cepstral-truncated 15-dimensional FDLP, indicating that increased spectral detail is useful even when supervision is provided indirectly at the word level. Our supervised variant of Laplacian eigenmaps posts significant gains over its unsupervised counterpart, but falls short of direct application of LDA and MLR to the reference vectors. This indicates that supervised discriminative training of a linear embedding is better than nonlinear embedding learned with implicit supervision. This suggests that discriminative nonlinear graph embedding techniques such as marginal Fisher analysis [31] may succeed in our setting. LDA applied to the output of unsupervised Laplacian eigenmaps outperforms LDA on its own, indicating that nonlinear graph embedding improves the linear separability of word types.

3.4. Discussion

Representative average precision scores for all of our methods are summarized in Table 3, organized according to the settings described in Section 2, along with the target dimensionalities that yielded the listed scores. For comparison, we include the setting in which an unsupervised Laplacian eigenmap embedding is learned from the test set (*UnsupTest*). This yields the best FDLP performance (0.416 AP) reported in this paper while using only $d = 20$ dimensions. Unfortunately, since it lacks an out-of-sample extension, this embedding is of limited practical utility.

Unsurprisingly, downsampling techniques, even nonuniform ones, fall short of the exhaustive alignment search performed under DTW. Embedding each speech segment with respect to a reference set encodes substantially more duration variability than downsampling, but still does not match the DTW baseline. PCA applied to reference vectors yields good word discriminability with fewer dimensions, but only with supervised embedding (LDA or MLR) do linear methods exceed the DTW baseline. Nonlinear embedding using Laplacian eigenmaps matches DTW using no supervision whatsoever, a significant result for zero-resource applications. Introducing supervision into this algorithm produces substantial gains, but falls short of the linear supervised embeddings produced by LDA and MLR. This indicates that nonlinearity is most important in the unsupervised setting. Combining Laplacian eigenmaps with LDA improves upon LDA alone, suggesting that Laplacian eigenmaps preserves or perhaps magnifies the information that makes LDA effective on its own. While different supervised methods produce the best performance at different operating points – the best performance on PLPs results from LDA applied to Laplacian eigenmaps while MLR posts the best FDLP results – the supervised methods all outperform the baselines and unsupervised methods.

Table 3. Representative average precision scores attained for each of the embedding schemes using r =10,000 reference examples (when applicable).

			Ave. Prec.	
Setting	**Algorithm**	d	**PLP**	**FDLP**
1. *NoTrain*	Baseline DTW	–	0.198	0.226
	Unif. Downsamp.	$25 \cdot p$	0.072	0.081
	Nonunif. ”	$25 \cdot p$	0.081	0.088
2. *UnsupTrain*	Ref. Vector	10,000	0.096	0.150
	PCA	200	0.081	0.139
	LapEig w/ OOS	200	0.195	0.236
3. *SupTrain*	Sup. LapEig	200	0.284	0.290
	LDA	50	0.346	0.293
	MLR	100	0.328	0.318
	LapEig + LDA	50	0.365	0.302
UnsupTest	Unsup. LapEig	20	0.253	0.416

Finally, the reference vectors required by some of our methods are expensive to construct. Table 2 shows that reference set size can be reduced with negligible loss in word discriminability. Figure 1(c) shows how reference set size affects task performance, with LDA target dimensionality chosen optimally for each condition. LDA beats the DTW baseline with as few as 1000 reference examples, a promising result, though the large gains in Table 3 require several thousand.

4. CONCLUSION

We have presented several fixed-dimensional embeddings of variable-length word segments appropriate for zero- and low-resource settings and investigated their performance on a word discrimination task. We find that with a limited unlabeled training set, unsupervised non-linear embeddings match the performance of a DTW baseline while embedding word segments in a space of 200 or fewer dimensions. When training data labels are known, we can greatly improve upon baseline performance with either linear (LDA, MLR) or nonlinear (supervised Laplacian eigenmaps) embeddings

of dimension 50-200. In some supervised and unsupervised training cases, we use reference vectors, which are related to Lipschitz embeddings, to define the initial fixed-dimensional space. Some natural directions for future work include selection of optimal reference sets, additional supervised distance learning approaches, and application of the ideas in downstream tasks such as query-by-example, spoken term discovery, and template-based speech recognition.

5. REFERENCES

[1] D. Gillick, L. Gillick, and S. Wegmann, "Don't multiply lightly: Quantifying problems with the acoustic model assumptions in speech recognition," in *Proc. ASRU*, 2011.

[2] T. N. Sainath et al., "Exemplar-based processing for speech recognition: An overview," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 98–113, 2012.

[3] De Wachter et al., "Template-based continuous speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1377–1390, 2007.

[4] G. Heigold, P. Nguyen, M. Weintraub, and V. Vanhoucke, "Investigations on exemplar-based features for speech recognition towards thousands of hours of unsupervised, noisy data," in *Proc. ICASSP*, 2012.

[5] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Tech. Rep., Michigan State Universiy, 2006.

[6] J. Labiak and K. Livescu, "Nearest neighbors with learned distances for phonetic frame classification," in *Proc. Interspeech*, 2011.

[7] A. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transcations on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[8] A. Jansen, K. Church, and H. Hermansky, "Towards spoken term discovery at scale with zero resources," in *Proc. Interspeech*, 2010.

[9] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. ASRU*, 2011.

[10] A. Jansen and B. Van Durme, "Indexing raw acoustic features for scalable zero resource search," in *Proc. Interspeech*, 2012.

[11] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. ASRU*, 2009.

[12] F. Metze et al., "The spoken web search task at MediaEval 2012," in *Proc. ICASSP*, 2013.

[13] V. Zue, J. Glass, M. Phillips, and S. Seneff, "The MIT SUMMIT speech recognition system: A progress report," in *Workshop on Speech and Natural Language*, 1989.

[14] J. R. Glass, "A probabilistic framework for segment-based speech recognition," *Speech Communication*, vol. 17, pp. 137–152, 2003.

[15] M. Ostendorf, "From HMMs to segment models: Stochastic modelling for CSR," in *Automatic Speech and Speaker Recognition: Advanced Topics (C.-H. Lee, F. K. Soong, and K. K. Paliwal, Eds.)*, chapter 8, pp. 185–209. Springer, 1996.

[16] O. Abdel-Hamid, L. Deng, D. Yu, and H. Jiang, "Deep segmental neural networks for speech recognition," in *Proc. Interspeech*, 2013.

[17] G. Zweig et al., "Speech recognition with segmental conditional random fields: A summary of the JHU CLSP 2010 summer workshop," in *Proc. ICASSP*, 2011.

[18] M. I. Layton and M. J. F. Gales, "Acoustic modelling using continuous rational kernels," in *Proc. MLSP*, 2005.

[19] A. Maas, S. Miller, T. O'Neil, A. Ng, and P. Nguyen, "Word-level acoustic modeling with convolutional vector regression," in *Proc. ICML Workshop on Representation Learning*, 2012.

[20] H. Tang, M. Hasegawa-Johnson, and T. Huang, "A novel vector representation of stochastic signals based on adapted ergodic HMMs," *IEEE Signal Processing Letters*, vol. 17, no. 8, pp. 715–718, 2010.

[21] G. R. Hjaltason and H. Samet, "Properties of embedding methods for similarity searching in metric spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 530–549, 2003.

[22] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. STOC*, 1998.

[23] G. Hristescu and M. Farach-Colton, "Cluster-preserving embedding of proteins," Tech. Rep. 99-50, Rutgers University, 1999.

[24] B. McFee and G. R. Lanckriet, "Metric learning to rank," in *Proc. ICML*, 2010.

[25] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, 2000.

[26] G. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *NIPS*, 2003.

[27] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Represenation," *Neural Computation*, vol. 16, pp. 1373–1396, 2003.

[28] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[29] V. S. Tomar and R. C. Rose, "Application of a locality preserving discriminant analysis approach to ASR," in *Proc. ISSPA*, 2012.

[30] D. Cai, J. Han, X. He, K. Zhou, and H. Bao, "Locality sensitive discriminant analysis," in *Proc. IJCAI*, 2007.

[31] S. Yan et al., "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[32] M. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Proc. ICASSP*, 2011.

[33] S. Thomas, S. Ganapathy, and H. Hermansky, "Phoneme recognition using spectral envelope and modulation frequency features," in *Proc. ICASSP*, 2009.

[34] A. Jansen et al., "A summary of the 2012 CLSP workshop on zero resource speech technologies and models of early language acquisition," in *Proc. ICASSP*, 2013.