Unsupervised Pattern Discovery in Speech

Alex S. Park, *Member, IEEE*, and James R. Glass, *Senior Member, IEEE*

***Abstract*—We present a novel approach to speech processing based on the principle of pattern discovery. Our work represents a departure from traditional models of speech recognition, where the end goal is to classify speech into categories defined by a prespecified inventory of lexical units (i.e., phones or words). Instead, we attempt to discover such an inventory in an unsupervised manner by exploiting the structure of repeating patterns within the speech signal. We show how pattern discovery can be used to automatically acquire lexical entities directly from an untranscribed audio stream. Our approach to unsupervised word acquisition utilizes a segmental variant of a widely used dynamic programming technique, which allows us to find matching acoustic patterns between spoken utterances. By aggregating information about these matching patterns across audio streams, we demonstrate how to group similar acoustic sequences together to form clusters corresponding to lexical entities such as words and short multiword phrases. On a corpus of academic lecture material, we demonstrate that clusters found using this technique exhibit high purity and that many of the corresponding lexical identities are relevant to the underlying audio stream.**

***Index Terms*—Speech processing, unsupervised pattern discovery, word acquisition.**

I. Introduction

OVER the last several decades, significant progress has been made in developing automatic speech recognition (ASR) systems which are now capable of performing large-vocabulary continuous speech recognition [1], [2]. In spite of this progress, the underlying paradigm of most approaches to speech recognition has remained the same. The problem is cast as one of classification, where input data (speech) is segmented and classified into a preexisting set of known categories (words). Discovering where these word entities come from is typically not addressed. This problem is of interest to us because it represents a key difference in the language processing strategies employed by humans and machines. Equally important, it raises the question of how much can be learned from speech data alone, in the absence of supervised input.

In this paper, we propose a computational technique for extracting words and linguistic entities from speech without supervision. The inspiration for our unsupervised approach to speech processing comes from two sources. The first source comes from a set of experiments conducted by developmental psychologists studying infant language learning. Saffran *et al.* found that 8-month-old infants are able to detect the statistical properties of commonly co-occurring syllable patterns, indicating that the identification of recurring patterns may be important in the word acquisition process [3]. Our second source of inspiration is implementational in nature and relates to current research in comparative genomics [4], [5]. In that area of research, pattern discovery algorithms are needed in order to find genes and structurally important sequences from massive amounts of genomic DNA or protein sequence data. Unlike speech, the lexicon of interesting subsequences is not known ahead of time, so these items must be discovered from the data directly. By aligning sequences to each other and identifying patterns that repeat with high recurrence, these biologically important sequences, which are more likely to be preserved, can be readily discovered. Our hope is to find analogous techniques for speech based on the observation that patterns of speech sounds are more likely to be consistent within word or phrase boundaries than across. By aligning continuous utterances to each other and finding similar sequences, we can potentially discover frequently occurring words with minimal knowledge of the underlying speech signal. The fundamental assumption of this approach is that acoustic speech data displays enough regularity to make finding such matches possible.

This paper primarily focuses on the unsupervised processing of speech data to automatically extract words and linguistic phrases. Our work differs substantially from other approaches to unsupervised word acquisition (see Section II) in that it operates directly on the acoustic signal, using no intermediate recognition stage to transform the audio into a symbolic representation. Although the inspiration for our methods is partially derived from experiments in developmental psychology, we make no claims on the cognitive plausibility of these word acquisition mechanisms in actual human language learning.

The results obtained in this paper are summarized as follows.

1) We demonstrate how to find subsequence alignments between the spectral representations of pairs of continuous utterances. In so doing, we propose a variation of a well-known dynamic programming technique for time series alignment, which we call segmental dynamic time warping (DTW). This task is motivated by the assumption that common words and phrases between utterance pairs are likely to be acoustically similar to each other. This algorithm allows us to find low distortion alignments between different regions of time in a given audio stream, which correspond to similar sounding speech patterns.
2) We show how low distortion alignments generated by the segmental DTW algorithm can be used to find recurring speech patterns in an audio stream. These patterns can be clustered together by representing the audio stream as an

Manuscript received January 10, 2007; revised July 26, 2007. This work was supported by the National Science Foundation under Grant #IIS-0415865. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Helen Meng.

A. S. Park was with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is now with Tower Research Capital, New York, NY 10013 USA (e-mail: malex@csail.mit.edu).

J. R. Glass is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jrg@csail.mit.edu).

Digital Object Identifier 10.1109/TASL.2007.909282

abstract adjacency graph. The speech pattern clusters that are discovered using this methodology are shown to correspond to words and phrases that are relevant to the audio streams from which they are extracted.

The remainder of this paper is organized as follows: We briefly survey related work in the areas of pattern discovery and unsupervised word acquisition in Section II. Section III describes the segmental DTW algorithm, an adaptation of a widely known dynamic programming technique, which is designed to find matching acoustic patterns between spoken utterances. In Section IV, we demonstrate how to induce a graph representation from the audio stream. We also employ clustering techniques to discover patterns that correspond to words and phrases in speech by aggregating the alignment paths that are produced by the segmental DTW algorithm. The experimental background for the experiments conducted in this paper are presented in Section V, including a description of the speech data used and specifics about our choice of signal representation. We give examples of the types of word entities found and analyze the results of our algorithm in Section VI, then conclude and discuss directions for future work in Section VII.

II. Related Work

There have been a variety of research efforts that are related to the work presented in this paper. We can roughly categorize these works into two major groups: applications of pattern discovery principles to domains outside of natural language processing, and unsupervised learning techniques within the field of natural language processing.

A. Pattern Discovery

The works summarized in this section represent a variety of different fields, ranging from computational biology to music analysis to multimedia summarization. There is a common underlying theme in all of this research: the application of pattern discovery principles to sequence data. We briefly describe work in each of these fields below.

In computational biology, research in pattern discovery algorithms is motivated by the problem of finding *motifs* (biologically significant recurring patterns) in biological sequences. Although the large body of proposed approaches is too large to list here, a survey of the more important techniques is described in [6] and [7]. The class of algorithms most relevant to our work are based upon sequence comparison, where multiple sequences are compared to one another to determine which regions of the sequence are recurring. Since biological sequences can be abstractly represented as strings of discrete symbols, many of the comparison techniques have roots in string alignment algorithms. In particular, a popular approach to alignment is the use of dynamic programming to search an edit distance matrix (also known as a distance matrix, position weight matrix, or position-specific scoring matrix) for optimal global alignments [8] or optimal local alignments [9]. The distance matrix is a structure which generates a distance or similarity score for each pair of symbols in the sequences being compared. We make use of distance matrices for alignment in this paper as well, although the sequences we work with are derived from the audio signal, and are therefore composed of real-valued vectors, not discrete symbols.

Distance matrices are also used extensively by researchers in the music analysis community. In this area of research, the music audio is parameterized as a sequence of feature vectors, and the resulting sequence is used to create a self-distance matrix. The structure of the distance matrix can then be processed to induce music structure (i.e., distinguish between chorus and verse), characterize musical themes, summarize music files, and detect duplicate music files [10]–[13]. We carry over the use of distance matrices for pattern discovery in music audio to our own work in speech processing.

B. Unsupervised Language Acquisition

The area of research most closely related to our work concerns the problem of unsupervised knowledge acquisition at the lexical level. Most recently, Roy *et al.* have proposed a model for lexical acquisition by machine using multimodal inputs, including speech [14]. Roy used a recurrent neural network trained on transcribed speech data to output a stream of phoneme probabilities for phonemically segmented audio. Words were learned by pairing audio and visual events and storing them as lexical items in a long-term memory structure.

In [15], de Marcken demonstrated how to learn words from phonetic transcriptions of continuous speech by using a model-based approach to lexicon induction. The algorithm iteratively updates parameters of the model (lexicon) to minimize the description length of the model given the available evidence (the input corpus).

Brent proposed a model-based dynamic programming approach to word acquisition by considering the problem as one of segmentation (i.e., inferring word boundaries in speech) [16]. In his approach, the input corpus is presented as a single unsegmented stream. The optimal segmentation of the corpus is found through a dynamic programming search, where an explicit probability model is used to evaluate each candidate segmentation. A similar strategy is used by Venkataraman in [17], although the utterance level representation of the corpus is used as a starting point rather than viewing the entire corpus as a single entity. The estimation of probabilities used in the segmentation algorithms of Brent and Venkataraman differ, but the overall strategies of the two techniques are conceptually similar. More recently, Goldwater has improved upon these model-based approaches by allowing for sparse solutions and more thoroughly investigating the role of search in determining the optimal segmentation of the corpus [18].

We note here that each of the above examples used a phonological lexicon as a foundation for the word acquisition process, and none of the techniques described were designed to be applied to the speech signal directly. The algorithms proposed by de Marcken and Roy, both depend on a phonetic recognition system to convert the continuous speech signal into a set of discrete units. The systems of Brent and Venkataraman were evaluated using speech data phonemically transcribed by humans in a way that applied a consistent phoneme sequence to a particular word entity, regardless of pronunciation.

Pattern discovery in audio has been previously proposed by several researchers. In [19], Johnson used a specialized distance metric for comparing covariance matrices of audio segments to find non-news events such as commercials and jingles

in broadcast news. Typically, the repeated events were identical to one another and were on the order of several seconds long. Unsupervised processing of speech has also been considered as a first step in acoustic model development [20]. Bacchiani proposed a method for breaking words into smaller acoustic segments and clustering those segments to jointly determine acoustic subword units and word pronunciations [21]. Similarily, Deligne demonstrated how to automatically derive an inventory of variable-length acoustic units directly from speech by quantizing the spectral observation vectors, counting symbol sequences that occur more than a specified number of times, and then iteratively refining the models that define each of these symbol sequences [22].

III. Segmental DTW

This section motivates and describes a dynamic programming algorithm which we call segmental DTW [23], [24]. Segmental DTW takes as input two continuous speech utterances and finds matching pairs of subsequences. This algorithm serves as the foundation for the pattern discovery methodology described in this paper.

Dynamic time warping was originally proposed as a way of comparing two whole word exemplars to each other by way of some optimal alignment. Given two utterances, $\mathcal{X}$ and $\mathcal{Y}$, we can represent each as a time series of spectral vectors, $(\mathbf{x}_1, \ldots, \mathbf{x}_{N_x})$ and $(\mathbf{y}_1, \ldots, \mathbf{y}_{N_y})$, respectively. The optimal alignment path between $\mathcal{X}$ and $\mathcal{Y}$, $\hat{\phi}$, can be computed, and the accumulated distortion between the two utterances along that path, $d_{\hat{\phi}}(\mathcal{X}, \mathcal{Y})$, can be used as a basis for comparison. Formally, we define a warping relation, or warp path, ϕ, to be an alignment which maps $\mathcal{X}$ to $\mathcal{Y}$ while obeying several constraints. The warping relation can be written as a sequence of ordered pairs

$$\phi = (i_k, j_k) \quad k = 1, \ldots, T \tag{1}$$

that represents the mapping

$$\mathbf{x}_{i_1} \leftrightarrow \mathbf{y}_{j_1}, \quad \mathbf{x}_{i_2} \leftrightarrow \mathbf{y}_{j_2}, \quad \ldots \quad \mathbf{x}_{i_N} \leftrightarrow \mathbf{y}_{j_N}.$$

In the case of global alignment, ϕ maps all of sequence $\mathcal{X}$ to all of sequence $\mathcal{Y}$. The globally optimal alignment is the one which minimizes

$$D_\phi(\mathcal{X}, \mathcal{Y}) = \sum_{k=1}^{T} d\left(\mathbf{x}_{i_k}, \mathbf{y}_{j_k}\right). \tag{2}$$

In (2), $d(\mathbf{x}, \mathbf{y})$ represents the unweighted Euclidean distance between feature vectors $\mathbf{x}$ and $\mathbf{y}$.

Although there are a number of spectral representations that are widely used in the speech research community, in this paper we use whitened Mel-scale cepstral coefficients (MFCCs). The process of whitening decorrelates the dimensions of the feature vector and normalizes the variance in each dimension. These characteristics of this spectral representation make the Euclidean distance metric a reasonable choice for comparing two feature vectors, as the distance in each dimension will also be uncorrelated and have equal variance. We note that our choice of feature representation and distance measure are not specifically designed to be stable when comparing different speakers or when comparing speech from different environmental conditions. Finding robust feature representations is a difficult problem in its own right, and we defer treatment of this issue to more extensive research done in the area.

When the utterances that we are trying to compare happen to be isolated words, the globally optimal alignment is a suitable way to directly measure the similarity of two utterances at the acoustic level. However, if the utterances consist of multiple words sequences, the distances and paths produced by optimal global alignment may not be meaningful. Although DTW was applied to the problem of connected word recognition via a framework called level building, this technique still required the existence of a set of isolated word reference templates [25]. In that respect, the problem has significant differences to the one in which we are interested. Consider the pair of utterances:

1) "He too was diagnosed with paranoid schizophrenia";
2) "... were willing to put Nash's schizophrenia on record."

Even in an optimal scenario, a global alignment between these two utterances would be forced to map speech frames from dissimilar words to one another, making the overall distortion difficult to interpret. This difficulty arises primarily because each utterance is composed of a different sequence of words, meaning that the utterances cannot be considered from a global perspective. However, (1) and (2) do share similarities at the local level. Namely, both utterances contain the word "schizophrenia." Identifying and aligning such similar local segments is the problem we seek to address in this section. Our proposed solution is a *segmental* variant of DTW that attempts to find subsequences of two utterances that align well to each other. Segmental DTW is comprised of two main components: a local alignment procedure which produces multiple warp paths that have limited temporal variation, and a path trimming procedure which retains only the lower distortion regions of an alignment path.

A. Local Alignment

In this section, we modify the basic DTW algorithm in several important ways. First, we incorporate global constraints to restrict the allowable shapes that a warp path can take. Second, we attempt to generate multiple alignment paths for the same two input sequences by employing different starting and ending points in the DTW search.

The need for global constraints in the DTW process can be seen by considering the example in Fig. 1. The shape of the path in the figure corresponds to an alignment that indicates that $\mathcal{X}$ is not a temporally dilated form of $\mathcal{Y}$, or vice versa. A more rigid alignment would prevent an overly large temporal skew between the two sequences, by keeping frames from one utterance from getting too far ahead of frames from the other. The following criterion, proposed by Sakoe, accomplishes this goal [26]. For a warp path, originating at (i_1, j_1), the kth coordinate of the path, $\mathcal{P}_k = (i_k, j_k)$, must satisfy

$$|(i_k - i_1) - (j_k - j_1)| \leq R. \tag{3}$$

The constraint in (3) essentially limits the path to a diagonal region of width $2R + 1$. This region is shown in Fig. 1, for a value of $R = 2$. Depending on the size of R, the ending point

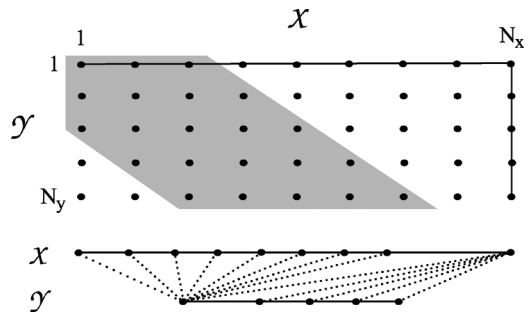


Fig. 1. Nonideal warp path that can result from unconstrained alignment. For this path, all frames from $\mathcal{X}$ are mapped to the first frame of $\mathcal{Y}$, and all frames from $\mathcal{Y}$ are mapped to the last frame of $\mathcal{X}$. The alignment corresponding to the warp path is displayed in the lower part of the figure. The shaded region of the graph represents the allowable set of path coordinates following the band constraint in (3) with $R = 2$.

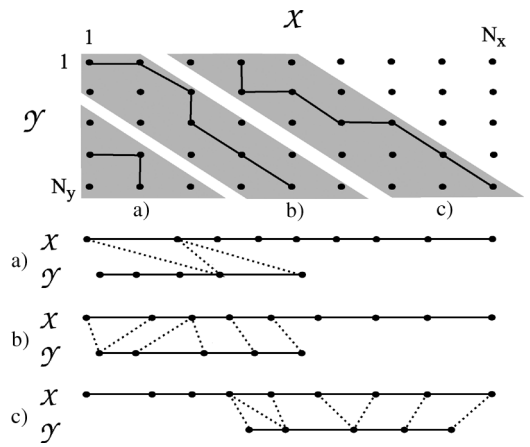


Fig. 2. Multiple alignment paths resulting from applying the band constraint with $R = 1$. The alignments corresponding to each diagonal region are shown below the grid.

of the constrained path may not reach (N_x, N_y). An alignment path resulting in unassigned frames in either of the input utterances may be desirable in cases where only part of the utterances match.

In addition to limiting temporal skew, the constraint in (3) also introduces a natural division of the search grid into regions suitable for generating multiple alignment paths with offset start coordinates as shown in Fig. 2.

For utterances of length N_x and N_y, with a constraint parameter of R, the start coordinates will be

$$((2R+1)k+1, 1), \quad 0 \le k \le \left\lfloor \frac{N_x - 1}{2R+1} \right\rfloor$$
$$(1, (2R+1)k+1), \quad 1 \le k \le \left\lfloor \frac{N_y - 1}{2R+1} \right\rfloor.$$

Based on these coordinates, we will have a number of diagonal regions, each defining a range of alignments between the two utterances with different offsets but the same temporal rigidity. Within each region, we can use dynamic time warping to find the optimal local alignment $\hat{\phi}_r$, where r is the index of the diagonal region.

B. Path Refinement

At this stage, we are left with a family of local warp paths $\hat{\phi}_r$ for $r = 1, \ldots, N_R$, where N_R is the number of diagonal regions. Because we are only interested in finding portions of the alignment which are similar to each other, the next step is to refine the warp path by discarding parts of the alignment with high distortion. Although there are a number of possible methods that could be used to accomplish this objective, we proceed by identifying and isolating the length-constrained minimum average (LCMA) distortion *fragment* of the local alignment path. We then *extend* the path fragment to include neighboring points falling below a particular threshold.

The problem of finding the LCMA distortion fragment can be described more generally as follows. Consider a sequence of positive real numbers

$$S = \langle s_1, \ldots, s_N \rangle \tag{4}$$

and a length constraint parameter L. Then, the length constrained minimum average subsequence LCMA(S, L) is a consecutive subsequence of S with length at least L that minimizes the average of the numbers in the subsequence. More formally, we wish to find m^* and n^* that satisfy

$$m^*, n^* = \underset{1 \le m \le n \le N}{\arg\min} \frac{1}{m-n} \sum_{k=m}^{n} s_k \tag{5}$$

with $n - m \ge L$. In our work, we make use of an algorithm proposed by Lin *et al.* for finding LCMA(S, L) in $O(N \log(L))$ time [27].

In order to apply this algorithm to our task, recall that every warp path ϕ is a sequence of ordered pairs

$$\phi = (i_1, j_1), \ldots, (i_T, j_T). \tag{6}$$

Associated with each warp path is a distortion sequence whose values are real and positive

$$\delta(\phi) = d(\mathbf{x}_{i_1}, \mathbf{y}_{j_1}), \ldots, d(\mathbf{x}_{i_T}, \mathbf{y}_{j_T}). \tag{7}$$

The minimum distortion warp path fragment φ is a subsequence of ϕ that satisfies

$$\delta(\varphi) = \text{LCMA}(\delta(\phi), L). \tag{8}$$

The minimum length criterion plays a practical role in computing the minimum average subsequence. Without the length constraint, the minimum average subsequence would typically be just the smallest single element in the original sequence. Likewise, for our application, it has the effect of preventing spurious matches between short segments within each utterance. The length criterion also has important conceptual implications. The value of L serves to control the granularity of repeating patterns that are returned by the segmental DTW procedure. Small values of L will lead to many short, subword patterns being found, while large values of L will return fewer, but more linguistically significant patterns such as words or phrases.

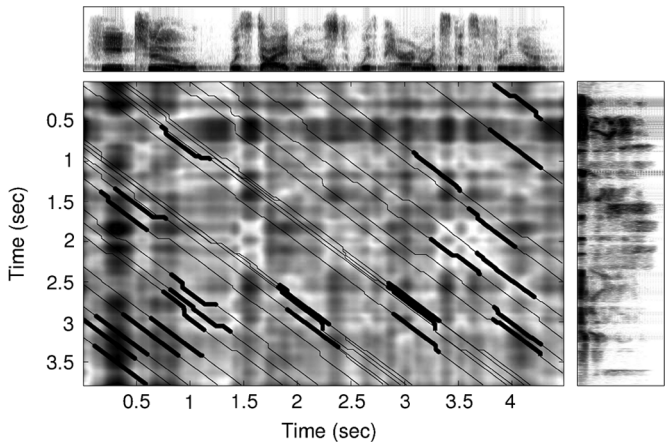


Fig. 3. Family of constrained warp paths $\hat{\phi}_r$ with $R = 10$ for the pair of utterances in our example. The frame rate for this distance matrix is 200 frames per second. The associated LCMA path fragments, with $L = 100$, are shown in bold as part of each warp path. Each path fragment is associated with an average distortion that indicates how well the aligned segments match one another.

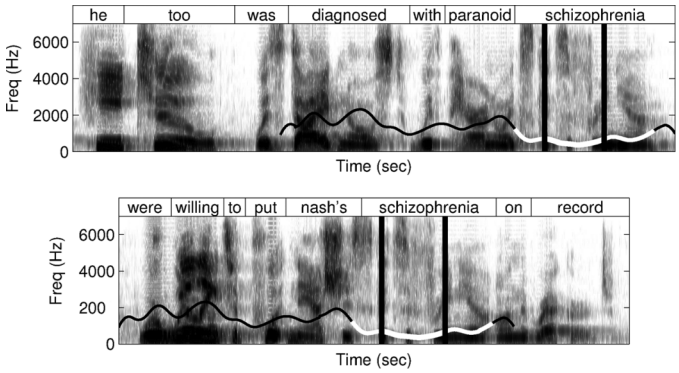


Fig. 4. Utterance level view of a warp path from Fig. 3. The segment bounded by the vertical black lines corresponds to the LCMA fragment for this particular warp path, while the remainder of the white line corresponds to the fragment resulting from extending the LCMA fragment to neighboring regions with low distortion.

In separate experiments, we found that the reliability of alignment paths found by the algorithm, in terms of matching accuracy, was positively correlated with path length [28]. This result, along with our need to limit the found paths to a manageable number for a given audio stream, led us to select a relatively long minimum length constraint of 500 ms. We discuss some less arbitrary methods for determining an optimal setting for L in Section VII. In the remainder of this section, we show example outputs that are produced when segmental DTW is applied to pairs of utterances.

C. Example Outputs

In this section, we step through the segmental DTW procedure for the pair of utterances presented at the beginning of Section III. The distance matrix for these two utterances is displayed in Fig. 3. In this distance matrix, each cell corresponds to the Euclidean distance between frames from each of the utterances being compared. The cell at row i, column j, corresponds to the distance between frame i of the first utterance and frame j of the second utterance. The local similarity between the utterance portions containing the word "schizophrenia" are evident in the diagonal band of low distortion cells stretching from the time coordinates (1.6, 0.9) to (2.1, 1.4). From the distance matrix, a family of constrained warp paths is found using dynamic time warping as shown in Fig. 3. The width parameter which constrains the extent of time warping is set to $R = 10$ frames, at a 5-ms analysis rate, which corresponds to a total allowable offset of 105 ms. The warp paths are overlaid with their associated length constrained minimum average path fragments. The length parameter used in this example is $L = 100$, which corresponds to approximately 500 ms. The coloring of the warp path fragments correspond to the average distortion of the path fragment, with bright red fragments indicating low distortion paths and darker shades indicating high distortion paths. Typically, there is a wide range of distortion values for the path fragments found, but only the lowest distortion fragments are of interest to us, as they indicate potential local matches between utterances.

An alternate view of the distortion path, including a frame-level view of the individual utterances, is shown in Fig. 4. This view of the distortion path highlights the need for extending the path fragments discovered using the LCMA algorithm. Although the distortion remains low from the onset of the word "schizophrenia" in each utterance, the LCMA path fragment (shown in red) starts almost 500 ms after this initial drop in distortion. In order to compensate for this phenomenon, we allow for path extension using a distortion threshold based on the values in the path fragment, for example within 10% of the distortion of the original fragment. The extension of the fragment is shown in Fig. 4 as a white line.

Although the endpoints of the extended path fragment in Fig. 4 happen to coincide with the common word boundaries for that particular example, in many cases, the segmental DTW algorithm will align subword sequences or even multiword sequences. This is because, aside from fragment length, the segmental DTW algorithm makes no use of lexical identity when searching for an alignment path.

IV. From Paths to Clusters

In order to apply the segmental DTW algorithm to an audio stream longer than a short sentence, we first perform silence detection on the audio stream to break it into shorter utterances. This segmentation step is described in more detail in Section V-B. Segmental DTW is then performed on each pair of utterances. With the appropriate choice of length constraint, this procedure generates a large number of alignment path fragments that are distributed throughout the original audio stream. Each alignment path consists of two intervals (the regions in time purported to be matching), and the associated distortion along that interval. Fig. 5 illustrates the distribution of path fragments throughout the audio stream. This visualization demonstrates how some time intervals in the audio match well to many other intervals, with up to 17 associated path fragments, while some time intervals have few, if any, matches. Since these fragments serve to link regions in time that are acoustically similar to one another, a natural question to ask is whether they can be used to build clusters of similar sounding speech segments with a common underlying lexical identity.

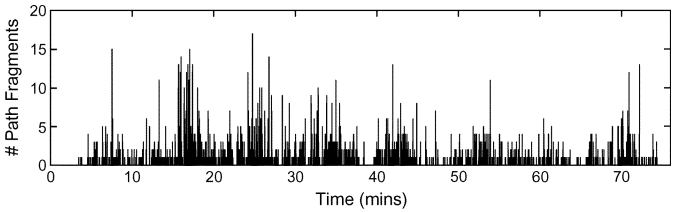


Fig. 5. Histogram indicating the number of path fragments present for each instant of time for the Friedman lecture. The distribution of path fragments is irregular, indicating that certain time intervals have more acoustic matches than others.

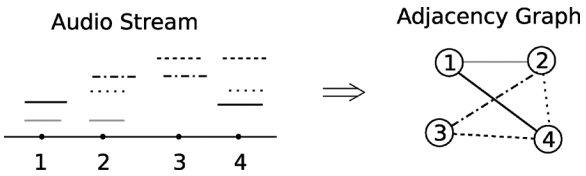


Fig. 6. Production of an adjacency graph from alignment paths and extracted nodes. The audio stream is shown as a timeline, while the alignment paths are shown as pairs of colored lines at the same height above the timeline. Node relations are captured by the graph on the right, with edge weights given by the path similarities.

Our approach to this problem is cast in a graph theoretical framework, which represents the audio stream as an abstract adjacency graph G consisting of a set of nodes V and a set of edges E. In this graph, the nodes correspond to locations in time, and the edges correspond to measures of similarity between those time indices. Given an appropriate choice of nodes and edges, graph clustering techniques can be applied to this abstract representation to group together the nodes in the graph that are closest to one another. Since graph clustering and partitioning algorithms are an active area of research [29]–[31], a wide range of techniques can be applied to this stage of the problem.

An overview of the graph conversion process is shown in Fig. 6. The time indices indicated in the audio stream are realized as nodes in the adjacency graph, while the alignment paths overlapping the time indices are realized as edges between the nodes. We use these alignment paths to derive edge weights by applying a simple linear transformation of the average path distortions, with the weight between two nodes being given by the following similarity score

$$e_{ij} = \mathcal{S}\left(\mathcal{P}(n_i, n_j)\right) = \frac{\theta - \mathcal{D}\left(\mathcal{P}(n_i, n_j)\right)}{\theta}. \tag{9}$$

In this equation, e_{ij} is the weight on the edge between nodes n_i and n_j, $\mathcal{P}(n_i, n_j)$ is the alignment path common to both nodes, $\mathcal{D}(\mathcal{P}(n_i, n_j))$ is the average distortion for that path, and θ is a threshold used to normalize the path distortions. The average distortion is used as opposed to the total distortion in order to normalize for path length when comparing paths with different durations. Paths with average distortion greater than θ are not included in the similarity computation. The distortion threshold chosen for all experiments in this chapter was 2.5, which resulted in approximately 10% of the generated alignment paths being retained. The resulting edge weights are closer to 1 between nodes with high similarity, and closer to zero (or nonexistent) for nodes with low similarity.

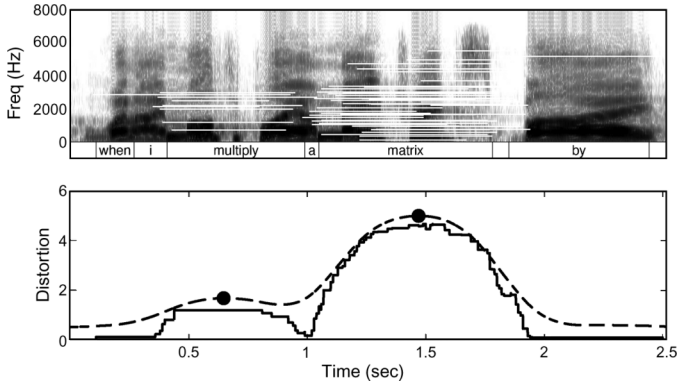


Fig. 7. Top—a partial utterance with the time regions from its associated path fragments shown in white. Paths are ordered from bottom to top in increasing order of distortion. Bottom—the corresponding similarity profile for the same time interval is shown as a solid line, with the smoothed version shown as a dashed line (raised for clarity). The extracted time indices are denoted as dots at the profile peaks.

A. Node Extraction

While it is relatively straightforward to see how alignment path fragments can be converted into graph edges given a set of time index nodes in the audio stream, it is less clear how these nodes can be extracted in the first place. In this section, we describe the node extraction procedure.

Recall that the input to the segmental DTW algorithm is not a single contiguous audio stream, but rather a set of utterances produced by segmenting the audio using silence detection. Our goal in node extraction is to determine a set of discrete time indices within these utterances that are representative of their surrounding time interval. This is accomplished by using information about the alignment paths that populate a particular utterance.

Consider the example shown in Fig. 7. In this example, there are a number of alignment paths distributed throughout the utterance with different average path distortions. The distribution of alignment paths is such that some time indices are covered by many more paths than others—and are therefore similar to more time indices in other utterances. These heavily covered time indices are typically located *within* the words and phrases that are matched via multiple alignment paths.

We can use the alignment paths to form a *similarity* profile by summing the similarity scores of (9) over time. That is, the similarity score at time t, is given by

$$S(t) = \sum_{\mathcal{P}, t \in \mathcal{P}} \mathcal{S}(\mathcal{P}). \tag{10}$$

In this equation, $\mathcal{P}$ are the paths that overlap t, and $\mathcal{S}(\mathcal{P})$ is the similarity value for $\mathcal{P}$ given by (9).

After smoothing the similarity profile with a 0.5-s triangular averaging window, we take the peaks from the resulting smoothed profile and use those time indices as the nodes in our adjacency graph. Because our extraction procedure finds locations with locally maximized similarity within the utterance, the resulting time indices demarcate locations that are more likely to bear resemblance to other locations in the audio stream.

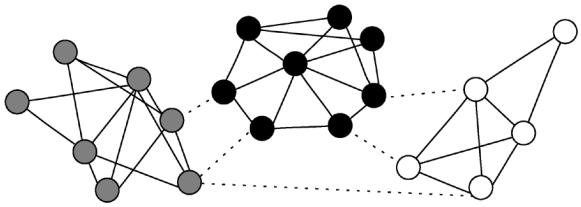


Fig. 8. Example of graph clustering output. Nodes are colored according to cluster membership. Dashed lines indicate intercluster edges.

The reasoning behind this procedure can be understood by noting that only some portions of the audio stream will have high similarity (i.e., low distortion) to other portions. By focusing on the peaks of the aggregated similarity profile, we restrict ourselves to finding those locations that are most similar to other locations. Since every alignment path covers only a portion of an utterance, the similarity profile will fluctuate over time. This causes each utterance to separate naturally into multiple nodes corresponding to distinct patterns that can be joined together via their common alignment paths. Each path that overlaps a node maps to an edge in the adjacency graph representation of the audio stream. The method we describe for inducing a graph from the alignment paths is one of many possible techniques. We discuss other possibilities for graph conversion in Section VII.

B. Graph Clustering

Once an adjacency graph has been generated for the audio stream using the extracted nodes and path fragment edges, the challenge of finding clusters in the graph remains. In an adjacency graph, a good clustering is one where nodes in one cluster are more densely connected to each other than they are to nodes in another cluster. The clustered adjacency graph in Fig. 8 illustrates this concept. A naive approach to this problem is to simply threshold the edge weights and use the groups of connected components that remain as clusters. Though conceptually simple, this approach is prone to accidental merging if even a single edge with high weight exists between two clusters that should be separated. In contrast to simple edge thresholding, a number of more sophisticated algorithms for automatic graph clustering have been proposed by researchers in other fields [32], [33]. For some applications, such as task scheduling for parallel computing, the clustering problem is cast as a partitioning task, where the number and size of desired clusters is known and the objective is to find the optimal set of clusters with those criteria in mind. For other applications, such as detecting community structure in social and biological networks, the number and size of clusters is typically unknown, and the goal is to discover communities and groups from the relationships between individuals.

In our work, the clustering paradigm aligns more closely with the latter example, as we are attempting to discover groups of segments corresponding to the same underlying lexical entity, and not partition the audio stream into a set of clusters with uniform size. Since a detailed treatment of the graph clustering problem is outside the scope and intent of this work, we focus on an efficient, bottom-up clustering algorithm for finding community structure in networks proposed by Newman [34]. The Newman algorithm begins with all edges removed and each node in its own group, then merges groups together in a greedy fashion by adding edges back to the graph in the order that maximizes a modularity measure Q which is given by

$$Q = \sum_i \left(e_{ii} - a_i^2\right) \tag{11}$$

where e_{ij} is the fraction of edges in the original network that connect vertices in group i to those in group j, and $a_i = \sum_j e_{ij}$. More informally, Q is the fraction of edges that fall within groups, minus the expected value of the same quantity if edges fall at random without regard for the community structure of the graph. The value of Q ranges between 0 and 1, with 0 being the expected modularity of a clustering where intercluster edges occured about as frequently as intracluster edges, and higher scores indicating more favorable clusterings of the graph. The advantages of this particular algorithm are threefold. First, it easily allows us to incorporate edge weight information in the clustering process by considering weights as fractional edges in computing edge counts. Second, it is extremely fast, operating in $O((V+E)V)$ time in the worst case. Finally, the modularity criterion offers a data-driven measure for determining the number of clusters to be detected from a particular graph.

Because our goal is to separate the graph into groups joining nodes sharing the same word(s), multiple groups containing the same word are more desirable than fewer groups containing many different words. We therefore associate a higher cost with the action of mistakenly joining two unlike groups than that of mistakenly leaving two like groups unmerged. This observation leads us to choose a conservative stopping point for the clustering algorithm at 80% of peak modularity.

C. Nodes to Intervals

Recall from Section IV-A that the nodes in the adjacency graph represent not time intervals in the original audio stream, but time indices. For the purposes of clustering, this time index abstraction may be adequate for representing nodes, but we will, at times, require associating a time interval corresponding to that node. One situation where we need a time interval rather than the time index corresponding to the node is for determining how to transcribe a node. As can be seen from the example in Fig. 7, the alignment paths overlapping a particular node rarely agree on starting and ending times for their respective time intervals. We assign a time interval to a node by computing the average start and end times for all the alignment paths for edges occurring within the cluster to which that node belongs.

V. Experimental Background

A. Speech Data

Word-level experiments in this paper are performed on speech data taken from an extensive corpus of academic lectures recorded at MIT [35]. At present, the lecture corpus includes more than 300 h of audio data recorded from eight different courses and over 80 seminars given on a variety of topics such as poetry, psychology, and science. Many of these lectures are publicly available on the MITWorld website [36] and as a part of the MIT OpenCourseware initiative [37]. In most cases, each lecture takes place in a classroom environment, and the

TABLE I
SEGMENT OF SPEECH TAKEN FROM A LECTURE, "THE WORLD IS FLAT," DELIVERED BY THOMAS FRIEDMAN

(1)	[um] I – you know – I don't know about the business side but she was very smart [um] about all of this and she said to me,
(2)	"You know Tom, everything we called the IT revolution? The information technology revolution, these last twenty years? Sorry to tell you, that was just the warm up act."
(3)	That has just been the sharpening, forging – forging, sharpening, and distribution of the tools of collaboration into this new platform

audio is recorded with an omnidirectional microphone (as part of a video recording).

We used six lectures for the experiments and examples in this work, each one delivered by a different speaker. The lectures ranged in duration from 47 to 85 min, with each focusing on a well-defined topic. Five of the lectures were academic in nature, covering topics like linear algebra, physics, and automatic speech recognition. The remaining lecture was delivered by Thomas Friedman, a New York Times columnist, who spoke for 75 min on the material in his recent book, "The World is Flat." An example of the type of speech from this lecture is shown in Table I. We note that transcript deviates significantly from patterns typically observed in formal written text, exhibiting artifacts such as filled pauses (1), false starts (1), sentence fragments (2), and sentence planning errors (3).

One of the unique characteristics of the lectures described above is the quantity of speech data that is available for any particular speaker. Unlike other sources of speech data, these lectures are primarily comprised of a single speaker addressing an audience for up to an hour or more at a time, making it particularly well suited to our word-discovery technique. Moreover, the focused and topical nature of the lectures we investigate tend to result in relatively small vocabularies which make frequent use of subject-specific keywords that may not be commonly used in everyday speech.

B. Segmentation

The lectures in the dataset are typically recorded as a single stream of audio often over 1 h in length, with no supplementary indicators of where one utterance stops and another begins. For many of the processing steps undertaken in subsequent stages, we require a set of discrete utterances in order to compare utterances to one another. In order to subdivide the audio stream into discrete segments of continuous speech, we use a basic phone recognizer to identify regions of silence in the signal [38]. Silent regions with duration longer than 2 s are removed, and the portions of speech in between those silences are used as the isolated utterances. The use of a phone recognizer is not a critical prerequisite for this segmentation procedure, since we only use the output to make a speech activity decision at each particular point in time. In the absence of a phone recognizer, a less sophisticated technique for speech activity detection can be substituted in its place. Most of the utterances produced during the segmentation procedure are short, averaging durations of less than 3 s. The segmentation procedure is also conservative enough that segmentation end points are rarely placed in the middle of a word.

TABLE II
CLUSTER STATISTICS FOR ALL LECTURES PROCESSED IN THIS PAPER. ONLY CLUSTERS WITH AT LEAST THREE MEMBERS ARE INCLUDED IN THIS TABLE. THE LAST TWO COLUMNS INDICATE HOW MANY OF THE GENERATED CLUSTERS ARE ASSOCIATED WITH A SINGLE WORD IDENTITY OR A MULTIWORD PHRASE

Lecture	Num Clusters	Avg Size	Avg Purity	# Single Word	#Multi Word
Friedman	63	5.59	79.63	25	31
ASR 2	92	7.36	86.13	44	44
ASR 6	87	10.05	93.22	47	40
ASR 19	63	10.32	91.85	33	29
Physics	51	10.39	89.46	31	20
Algebra	41	8.80	93.98	30	11

C. Computational Considerations

As described, the pattern discovery process requires that each utterance is compared with each other utterance. The number of segmental DTW comparisons required for each audio stream is therefore quadratic in the number of utterances. This step is the most computationally intensive part of the process; node generation and clustering do not incur significant computation costs. Since each pair of utterances can be compared independently, we perform these comparisons in parallel to speed up computation. The number of comparisons can potentially be reduced by merging matching segments as they are found.

VI. CLUSTER ANALYSIS

We processed the six lectures described in Section V-A using the segmental DTW algorithm and generated clusters for each. Overall cluster statistics for these lectures are shown in Table II. We will return to this table momentarily, but for illustrative purposes, we focus on clusters from the Thomas Friedman lecture. A more detailed view of the clusters with at least three members is shown in Table III. In this table, the clusters are listed first in decreasing order of size, denoted by $|\mathcal{C}|$, then by decreasing order of density $D(\mathcal{C})$. The density, a measure of the "interconnectedness" of each cluster, is given by

$$D(\mathcal{C}) = \binom{|\mathcal{C}|}{2}^{-1} \sum_{n_1,n_2 \in \mathcal{C}} w(n_1, n_2). \tag{12}$$

The quantity in the above equation is the fraction of edges observed in the cluster out of all possible edges that could exist between cluster nodes. Higher densities indicate greater agreement between nodes. Table III also includes a purity score for each cluster. The purity score is a measure of how accurately the clustering algorithm is able to group together like speech nodes, and is determined by calculating the percentage of nodes that agree with the lexical identity of the cluster. The cluster identity, in turn, is derived by looking at the underlying reference transcription for each node and choosing the word or phrase that appears most frequently in the nodes of that particular cluster. Clusters with no majority word or phrase (such as those matching subword speech segments), are labeled as "–."

1) Example Clusters: Some examples of specific clusters with high purity are shown in Fig. 9. Cluster 27 in Fig. 9 is an

TABLE III
INFORMATION FOR THE 63 CLUSTERS WITH AT LEAST THREE MEMBERS GENERATED FOR THE FRIEDMAN LECTURE. CLUSTERS ARE ORDERED FIRST BY SIZE, THEN IN DECREASING ORDER OF DENSITY

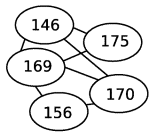
$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	Transcription	Purity	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	Transcription	Purity	$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$D(\mathcal{C})$	Transcription	Purity
1	27	0.017	applications	29.6	23	5	0.043	governance	40.0	43	3	0.110	system	100.0
2	25	0.020	collaboration	88.0	24	4	0.169	search engine optimizer	100.0	44	3	0.107	fourteen ninety two	100.0
3	22	0.043	globalization	100.0										
4	21	0.023	imagination	38.1	25	4	0.169	toshiba laptop	100.0	45	3	0.106	solar powered	66.7
5	12	0.032	platform	100.0	26	4	0.146	work together	100.0	46	3	0.102	japanese	100.0
6	12	0.023	flattener	75.0	27	4	0.105	inflection point	75.0	47	3	0.101	internet	100.0
7	8	0.053	fiberoptic cable	50.0	28	4	0.070	haven connecticut	50.0	48	3	0.093	-	0.0
8	8	0.030	to ups	37.5	29	4	0.066	flatten the world	100.0	49	3	0.090	imported	100.0
9	7	0.033	southwest airlines	100.0	30	4	0.059	economic playing field	100.0	50	3	0.087	um	100.0
										51	3	0.083	quiet crisis	100.0
10	6	0.102	(n)ever before	100.0	31	4	0.053	ten percent	50.0	52	3	0.081	discover	100.0
11	6	0.098	informing	100.0	32	4	0.053	the world	100.0	53	3	0.075	-	0.0
12	6	0.077	the history of	83.3	33	4	0.050	more people	100.0	54	3	0.071	standards	66.7
13	6	0.073	flat world	100.0	34	4	0.047	-	0.0	55	3	0.069	governance	66.7
14	6	0.065	outsourcing	100.0	35	4	0.046	supply chain	100.0	56	3	0.066	-	0.0
15	6	0.036	-	0.0	36	4	0.040	-	0.0	57	3	0.065	business processes	100.0
16	6	0.030	the beginning	33.3	37	3	0.203	plug and play	100.0	58	3	0.064	software	100.0
17	5	0.135	globalizing	100.0	38	3	0.201	knowledge and work	100.0	59	3	0.064	the night before	100.0
18	5	0.111	open source	100.0						60	3	0.064	the world is flat	100.0
19	5	0.106	individuals	100.0	39	3	0.142	china	100.0	61	3	0.059	u p s	66.7
20	5	0.080	two thousand	100.0	40	3	0.139	the berlin wall	100.0	62	3	0.053	productivity	100.0
21	5	0.076	reservation	100.0	41	3	0.133	multinationals	100.0				boost	
22	5	0.054	horizontal	100.0	42	3	0.114	huge	100.0	63	3	0.050	-	100.0

example of a high-density cluster, with each node connecting to each other node, and the underlying transcriptions confirm that each node corresponds to the same recurring phrase. The other two clusters in Fig. 9, while not displaying the same degree of interconnectedness, nevertheless all consist of nodes with similar transcriptions. One interesting property of these clusters is the high degree of temporal locality displayed by their constituent nodes. With the exception of node 587, most of the other nodes occur within 5 min of the other nodes in their respective clusters. This locality may be indicative of transient topics in the lecture which require the usage of terms that are only sporadically used. In the case of cluster 27, these four instances of "search engine optimize-" were the only instances where they were spoken in the lecture.

2) Cluster Statistics: Several interesting points can be noted regarding the clusters generated from the Friedman lecture. First, most clusters (56 of 63) have a word or phrase that can be considered to be the lexical identity of the cluster. Out of these clusters, over 73% of the clusters have a purity of 100%, which offers encouraging evidence that the segmental DTW measures and subsequent clustering procedure are able to correctly group recurring words together. As might be expected, the cluster density appears to be positively correlated to cluster purity, with an average purity of 87% among clusters with density greater than 0.05, and an average purity of 53% among clusters with density less than or equal to 0.05. We also observe that the clustering algorithm does not appear to discriminate between

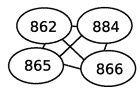
Node Index	Time Index	Transcription
146	16:03	countries globalizing
156	16:44	companies globalizing
169	17:27	countries globalizing
170	17:30	companies globalizing
175	17:40	globalizing

(a)



Node Index	Time Index	Transcription
862	1:06:14	a search engine optimizer
865	1:06:22	be a search engine optimizer
866	1:06:32	search engine optimizing
884	1:08:45	a search engine optimizing

(b)



Node Index	Time Index	Transcription
239	21:08	than ever before
272	23:27	than ever before in
295	24:41	ever before
302	25:08	never before
308	25:15	ever before
587	42:34	than ever before

(c)

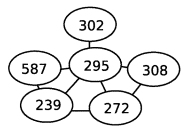


Fig. 9. Detailed view of clusters 17, 24, and 10, including the node indices, transcriptions, and locations in the audio stream.

TABLE IV
TWENTY MOST RELEVANT WORDS FOR EACH LECTURE, LISTED IN DECREASING ORDER OF TFIDF SCORE. WORDS OCCURING AS PART OF A CLUSTER FOR THAT LECTURE ARE SHOWN IN BOLD

Friedman	Algebra	Physics	ASR L2	ASR L6	ASR L19
flat	**matrix**	**electric**	**frequency**	**cluster**	**speaker**
globalization	**row**	zero	**vocal**	**distortion**	**adaptation**
collaboration	**zero**	**sphere**	**wave**	**data**	**model**
india	**pivot**	**charge**	**transform**	algorithm	**vector**
era	**equation**	**plate**	**fourier**	metric	**parameter**
flattener	**elimination**	**symmetry**	vowel	**vector**	adapt
dollar	**column**	**flux**	**speech**	**distance**	**technique**
china	**multiply**	**plane**	**cavity**	**speech**	**utterance**
southwest	**matrices**	vector	signal	split	weight
argue	**subtract**	**uniformly**	tract	**assign**	**likelihood**
airline	**minus**	**gauss**	**fold**	**quantization**	**estimate**
thousand	step	**field**	**sound**	**dimension**	dependent
outsourcing	**multiplication**	**angle**	**acoustic**	**train**	independent
really	**exchange**	**epsilon**	**window**	**iteration**	**data**
platform	**inverse**	divided	**characteristic**	**plot**	**recognize**
huge	**suppose**	**vandegraaff**	**function**	**coefficient**	**speech**
create	**plus**	**surface**	**source**	**mean**	error
convergence	negative	**distribute**	**velocity**	pick	**cluster**
connect	**substitution**	**inside**	tongue	**merge**	mean
chapter	identity	**sigma**	noise	**criterion**	filter

single words and multiword phrases that are frequently spoken as a single entity, with more than half of the clusters (31 of 56) mapping to multiword phrases.

Overall cluster purity statistics for the five other academic lecture processed in this paper are shown in Table II. We found that across all six lectures, approximately 83% of the generated clusters had density greater than 0.05, and among these higher density clusters, the average purity was 92.2%. In contrast, the average purity across all of the lower density clusters was only 72.6%. These statistics indicate that the observations noted in the previous paragraph appear to transfer to the other lectures. Some notable differences between the Friedman lecture and the academic lectures are the larger average cluster size, and higher overall purity across the clusters in general. The larger size of some clusters can be attributed to the more focused nature of the academic lecture vocabulary, while the higher purity may be a result of differences in speaking style.

A cursory view of the cluster identities for each lecture indicates that many clusters correspond to words or phrases that are highly specific to the subject material of that particular lecture. For example, in the physics lecture, the words "charge," "electric," "surface," and "epsilon," all correspond to some of the larger clusters for the lecture. This phenomenon is expected, since relevant content words are likely to recur more often, and function words such as "the," "is," and "of," are of short duration and typically exhibit significant pronunciation variation as a result of coarticulation with adjacent words. One way of evaluating how well the clusters capture the subject content of a lecture is to consider the coverage of relevant words by the generated clusters.

Since there is no easy way of measuring word relevancy directly, for the purposes of our work, we use each word's term-frequency, inverse document-frequency (TFIDF) score as a proxy for its degree of relevance [39]. The TFIDF score is the frequency of the word within a document normalized by the frequency of the same word across multiple documents. Our rationale for using this score is that words with high frequency within the lecture, but low frequency in general usage are more likely to be specific to the subject content for that lecture. The word lists in Table IV are the 20 most relevant words for each lecture ranked in decreasing order of their TFIDF score. Each list was generated as follows.

1) First, words in the reference transcription were stemmed to merge pluralized nouns with their root nouns, and various verb tenses with their associated root verbs.
2) Partial words, filled pauses, single letters and numbers, and contractions such as "you've" or "i'm" were removed from the reference transcription.
3) Finally, the remaining words in the lecture were ranked by TFIDF, where the document frequency was determined using the 2K most common words in the Brown corpus [40].

When considered in the context of each lecture's title, the lists of words generated in Table IV appear to be very relevant to the subject matter of each lecture, which qualitatively validates our use of the TFIDF measure. The words for each lecture in Table IV are highlighted according to their cluster coverage, with words represented by a cluster shown in bold. On average, 14.8 of the top 20 most relevant words are covered by a cluster generated by our procedure. This statistic offers encouraging

evidence that the recurring acoustic patterns discovered by our approach are not only similar to each other (as shown by the high average purity), but also informative about the lexical content of the audio stream.

VII. Discussion and Future Work

This paper has focused on the unsupervised acquisition of lexical entities from the information produced by the segmental DTW algorithm. We demonstrated how to use alignment paths, which indicate pairwise similarity, to transform the audio stream into an abstract adjacency graph which can then be clustered using standard graph clustering techniques. As part of our evaluation, we showed that the clusters generated by our proposed procedure have both high purity and good coverage of terms that are relevant to the subject of the underlying lecture.

As we noted in Section V-A, there are several reasons why the lecture audio data was particularly well suited for pattern discovery using segmental DTW. First, the types of material was single-speaker data in a consistent environment, which allowed us to ignore issues of robustness with our feature representation. Second, the amount of topic-specific data ensured that there were enough instances of repeated words for the algorithm to find. For both of these reasons, our algorithm would likely not perform as well if applied directly to other domains, such as Switchboard or Broadcast News. In particular, we would not expect to find clusters of the same size or density without reducing the length parameter and/or including more edges in the adjacency graph. The reason for this is mainly due to speaker changes and paucity of repeated content words. Speaking style is not as significant an issue, as the lecture data exhibits speech that is much more conversational than read speech or broadcast news data.

The work presented in this paper represents only an initial investigation into the more general problem of knowledge acquisition from speech. Many directions for future work remain, and we expand upon some of them here.

In our experiments, we chose a large value for the L parameter to limit the over-generation of alignment path fragments corresponding to short, possibly spurious, acoustic matches. Typically, low-distortion path fragments corresponding to words or phrases are recoverable from shorter path fragments during the extension step of path refinement. Discovery of longer fragments is therefore not particularly sensitive to our choice of L. Larger values of L primarily serve to prevent short path fragments (usually corresponding to subword matches) from being passed on to the node generation and clustering stage. Within the context of word acquisition, these shorter path fragments are problematic because they cause dissimilar words to cluster with one another via common subword units. Possibilities for future work include using smaller values of L for discovery of subword units or determining the appropriate setting of L in a more principled manner. For example, the optimal setting for L could be determined by performing pattern discovery over the audio stream using multiple L's and choosing from the best one according to some selection criterion.

An incremental strategy for improving cluster purity and finding more precise word boundaries may be to adopt an iterative approach to cluster formation. After clusters have been formed and the time intervals for each node have been estimated, edge weights between cluster nodes can be recomputed using the start and end times of the node intervals as constraints. Based on these new edge weights, nodes can be rejected from the cluster and the time intervals can be reestimated, with the process continuing until convergence to a final set of nodes. The idea behind this approach is to eliminate chaining and partial match errors by forcing clusters to be generated based on distortions that are computed over a consistent set of speech intervals.

Similarly, one could imagine using an interval-based clustering strategy to help avoid accidental merging of lexically different clusters, which can occur as a result of "chained" multiword phrases, or matched subword units such as "tion." Interval-based clustering would resolve this problem by using whole time intervals as nodes, rather than time indices. This approach would allow a hierarchical representation of a particular speech segment and distinguish between overlapping intervals of different lengths.

At a more abstract level, we believe that an interesting direction for future work would be to incorporate some way to build and update a model of the clustered intervals using some type of hidden Markov model or generalized word template. This would introduce significant computational savings by reducing the number of required comparisons.

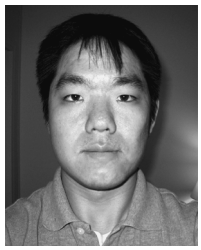
Another area for future exploration is the automatic identification and transcription of cluster identities. We have previously proposed algorithms for doing so using isolated word recognition and phonetic recognition combined with a large baseform dictionary [24]. This task illustrates how unsupervised pattern discovery can provide complementary information to more traditional automatic speech recognition systems. Since most speech recognizers process each utterance independently of one another, they typically do not take advantage of the consistency with which the same word is uttered when repeated in the test data. Alignment paths generated by segmental DTW can find locations where an automatic transcription is not consistent by indicating where acoustically similar segments produced different transcriptions.

This paper documents our initial research on unsupervised strategies for speech processing. While conventional large vocabulary speech recognition would likely perform well in matched training and testing scenarios, there are many real-world scenarios where a paucity of content information can expose the brittleness of purely supervised approaches. We believe that techniques such as the one in this paper, which rely less on training data, can be combined with conventional speech recognizers to create more flexible, hybrid systems that can learn from and adapt to unexpected input. Examples of such unexpected input include: accented speech, out-of-vocabulary words, new languages, and novel word usage patterns. In each of these scenarios, exploiting the consistency of repeated patterns in the test data has not been fully explored, and we believe it is a promising direction for future research.

References

[1] J. L. Gauvain, L. Lamel, and G. Adda, "The LIMSI broadcast news transcription system," *Speech Commun.*, vol. 37, no. 1–2, pp. 89–108, May 2002.

[2] A. Ljolje, D. M. Hindle, M. D. Riley, and R. W. Sproat, "The AT&T LVCSR-2000 system," in *Proc. DARPA Speech Transcription Workshop*, College Park, MD, May 2000 [Online]. Available: http://www.nist.gov/speech/publications/tw00/pdf/cts30.pdf
[3] J. R. Saffran, R. N. Aslin, and E. L. Newport, "Statistical learning by 8-month old infants," *Science*, vol. 274, pp. 1926–1928, Dec. 1996.
[4] I. Rigoutsos and A. Floratos, "Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm," *Bioinformatics*, vol. 14, no. 1, pp. 55–67, Feb. 1998.
[5] A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert, "Approaches to the automatic discovery of patterns in biosequences," *J. Comput. Biol.*, vol. 5, no. 2, pp. 279–305, 1998.
[6] G. K. Sandve and F. Drabløs, "A survey of motif discovery methods in an integrated framework," *Biol. Direct*, vol. 1, pp. 1–11, Apr. 2006.
[7] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge, U.K.: Cambridge Univ. Press, 1998.
[8] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, pp. 443–453, 1970.
[9] M. S. Waterman and M. Eggert, "A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons," *J. Mol. Biol.*, vol. 197, pp. 723–725, 1987.
[10] B. Logan and S. Chu, "Music summarization using key phrases," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, Jun. 2000, pp. 749–752.
[11] C. J. Burges, D. Plastina, J. C. Platt, E. Renshaw, and H. Malvar, "Using audio fingerprinting for duplicate detection and thumbnail generation," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Philadelphia, PA, Mar. 2005, vol. 3, pp. 9–12.
[12] R. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," in *Proc. Int. Conf. Music Inf. Retrieval*, Paris, France, Oct. 2002, pp. 63–70.
[13] M. Goto, "A chorus-section detecting method for musical audio signals," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2003, vol. 5, pp. 437–440.
[14] D. Roy and A. Pentland, "Learning words from sights and sounds: A computational model," *Cognitive Sci.*, vol. 26, no. 1, pp. 113–146, Jan. 2002.
[15] C. G. de Marcken, "Unsupervised language acquisition," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1996.
[16] M. R. Brent, "An efficient probabilistically sound algorithm for segmentation and word discovery," *Mach. Learn.*, vol. 34, no. 1–3, pp. 71–105, Feb. 1999.
[17] A. Venkataraman, "A statistical model for word discovery in transcribed speech," *Comput. Ling.*, vol. 27, no. 3, pp. 352–372, Sep. 2001.
[18] S. Goldwater, T. Griffiths, and M. Johnson, "Contextual dependencies in unsupervised word segmentation," in *Proc. Coling/ACL*, Sydney, Australia, 2006, pp. 673–670.
[19] S. Johnson and P. Woodland, "A method for direct audio search with application to indexing and retrieval," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, 2000, pp. 1427–1430.
[20] J. Glass, "Finding acoustic regularities in speech: Application to phonetic recognition," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 1988.
[21] M. Bacchiani and M. Ostendorf, "Joint lexicon, acoustic unit inventory and model design," *Speech Commun.*, vol. 29, no. 2–4, pp. 99–114, Nov. 1999.
[22] S. Deligne and F. Bimbot, "Inference of variable length acoustic units for continuous speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Munich, Germany, 1997, vol. 3, pp. 1731–1734.
[23] A. Park and J. Glass, "Towards unsupervised pattern discovery in speech," in *Proc. IEEE Workshop Autom. Speech Recognition Understanding*, San Juan, Puerto Rico, 2005, pp. 53–58.
[24] A. Park and J. R. Glass, "Unsupervised word acquisition from speech using pattern discovery," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, Apr. 2006, pp. I-409–I-412.
[25] C. S. Myers and L. R. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *IEEE Trans. Acoust, Speech, Signal Process.*, vol. ASSP-29, no. 2, pp. 284–297, Apr. 1981.
[26] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.
[27] Y.-L. Lin, T. Jiang, and K.-M. Chao, "Efficient algorithms for locating the length-constrained heaviest segments with applications to biomolecular sequence analysis," *J. Comput. Syst. Sci.*, vol. 65, no. 3, pp. 570–586, Jan. 2002.
[28] A. Park, "Unsupervised pattern discovery in speech: Applications to word acquisition and speaker segmentation," Pd.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1988.
[29] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 22, no. 8, pp. 888–905, Aug. 2000 [Online]. Available: citeseer.ist.psu.edu/article/shi97normalized.html.
[30] M. Meila and J. Shi, "Learning segmentation by random walks," in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Cambridge, MA: MIT Press, 2001, vol. 13, pp. 873–879.
[31] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 849–856.
[32] S. White and P. Smyth, "A spectral clustering approach to finding communities in graphs," in *SIAM Int. Conf. Data Mining*, Newport Beach, CA, 2005, pp. 274–285.
[33] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, 2004, 026113.
[34] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, 2004, 066133.
[35] J. Glass, T. J. Hazen, L. Hetherington, and C. Wang, "Analysis and processing of lecture audio data: Preliminary Investigations," in *Proc. HLT-NAACL 2004 Workshop Interdisciplinary Approaches to Speech Indexing and Retrieval*, Boston, MA, May 2004, pp. 9–12.
[36] MIT, MIT World [Online]. Available: http://mitworld.mit.edu.
[37] MIT, "MIT Open Courseware," [Online]. Available: http://ocw.mit.edu.
[38] J. Glass, "A probabilistic framework for segment-based speech recognition," *Comput. Speech Lang.*, vol. 17, no. 2–3, pp. 137–152, 2003.
[39] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," Cornell Univ., Ithaca, NY, Tech. Rep. TR87-881, 1987.
[40] W. N. Francis and H. Kucera, *Frequency Analysis of English Usage: Lexicon and Grammar*. Boston, MA: Houghton-Mifflin, 1982.



Alex S. Park (M'06) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2001, 2002, and 2006, respectively.

While at MIT, he performed his doctoral research as a member of the Spoken Language Systems Group in the Computer Science and Artificial Intelligence Laboratory. His research interests include unsupervised learning in speech, auditory signal processing, speaker recognition, and noise robust speech recognition. While a student, he took part in research internships with Nuance Communications and ATR Laboratories. He is currently with Tower Research Capital in New York.

James R. Glass (SM'06) received the S.M. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1985, and 1988, respectively.

After starting in the Speech Communication Group at the MIT Research Laboratory of Electronics, he has worked at the Laboratory for Computer Science, now the Computer Science and Artificial Intelligence Laboratory (CSAIL), since 1989. Currently, he is a Principal Research Scientist at CSAIL, where he heads the Spoken Language Systems Group. He is also a Lecturer in the Harvard-MIT Division of Health Sciences and Technology. His primary research interests are in the area of speech communication and human–computer interaction, centered on automatic speech recognition and spoken language understanding. He has lectured, taught courses, supervised students, and published extensively in these areas.

Dr. Glass has been a member of the IEEE Signal Processing Society Speech Technical Committee, and an Associate Editor for the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.