Pattern Recognition Letters
journal homepage: www.elsevier.com

Assisted keyword indexing for lecture videos using unsupervised keyword spotting

Manish Kanadje[a,**], Zachary Miller[a], Anurag Agarwal[b], Roger Gaborski[a], Richard Zanibbi[a], Stephanie Ludi[c]

[a]*Department of Computer Science,*
[b]*School of Mathematical Sciences,*
[c]*Software Engineering Department,*
Rochester Institute of Technology, 102 Lomb Memorial Drive, Rochester, NY 14623-5608, United States of America

ABSTRACT

Many students use videos to supplement learning outside the classroom. This is particularly important for students with challenged visual capacities, for whom seeing the board during lecture is difficult. For these students, we believe that recording the lectures they attend and providing effective video indexing and search tools will make it easier for them to learn course subject matter at their own pace. As a first step in this direction, we seek to help instructors create an index for their lecture videos using audio keyword search, with queries recorded by the instructor on their laptop and/or created from video excerpts. For this we have created an unsupervised within-speaker keyword spotting system. We represent audio data using de-noised, whitened and scale-normalized Mel Frequency Cepstral Coefficient (MFCC) features, and locate queries using Segmental Dynamic Time Warping (SDTW) of feature sequences. Our system is evaluated using introductory Linear Algebra lectures from instructors with different accents at two U.S. universities. For lectures produced using a video camera at RIT, laptop-recorded queries obtain an average Precision at 10 of 71.5%, while 79.5% is obtained for within-lecture queries. For lectures recorded using a lapel microphone at MIT, using a similar keyword set we obtain a much higher average Precision at 10 of 89.5%. Our results suggest that our system is robust to changes in environment, speaker and recording setup.

1. Introduction

In recent times, there has been a significant increase in digital content in order to supplement the learning of students. Video recordings of classroom lectures can help students to improve their understanding significantly. With video recordings, students may access lecture content multiple times according to their need. However, video lectures do not have a well-defined index. Students have to manually search to reach a point of interest. This is a tedious task. However, this task becomes increasingly difficult for people having challenges in visual capacities. A text-like index for the video content will be immensely helpful for such students, in order to improve the accessibility of video lectures.

AccessMath is a video lecture indexing and retrieval system being designed at our institute. The main goal of AccessMath system is to facilitate the learning of linear algebra lectures for students having challenges in visual capabilities. This paper describes the audio indexing portion of AccessMath. We plan for AccessMath to eventually be a lecture indexing and retrieval system accepting queries issued in image, audio or text formats. Using this system, a student could search a linear algebra lecture for a formula, e.g. $A\bar{x} = \bar{b}$, by selecting a part of an image or a spoken query from the lecture.

We propose a keyword spotting system which will enable an instructor or student to perform search using audio queries spoken by the instructor. We have also created a prototype to help instructors and students organize search hits generated by the system. This system helps create an index similar to the table of contents for a textbook using within-speaker audio queries. Keyword spotting is a relatively difficult task as differences in speech characteristics such as accent, pitch and environment cause high variance in utterances of the same keywords. In the proposed system, we have considered single channel audio input created in a single speaker environment.

**corresponding author:
e-mail: mk2852@rit.edu (Manish Kanadje)

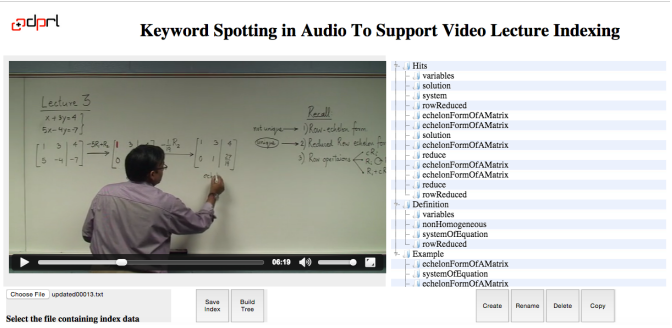

Fig. 1. Using indexing tools the user can play the video lecture from the point of generated hits. The tree based indexing structure helps the user to organize hits into groups such as 'Definition' or 'Example'.

Keyword spotting systems convert an input speech signal into a temporal spectral vector. After modeling the speech signal, these systems usually fall within two different categories: Dynamic Time Warping-based or Hidden Markov Model-based. Dynamic Time Warping (DTW) finds an optimal alignment between two audio sequences, seeking to determine whether they represent the same word (Rabiner et al., 1978). DTW matches two temporal sequences by non-linearly comparing audio frames and calculating the cost of alignment. In contrast, Hidden Markov Model-based approaches require training data for creating probabilistic temporal models for individual words. DTW does not require labeled data for training. However, the cost of computation is high for DTW, $O(mn)$ where m and n are sequence lengths. For this reason, many variations of DTW attempt to reduce its computational cost.

As shown in Figure 1, our system creates an index of candidates for a query within a lecture.[1] We have offered the functionality of hierarchical annotations to make this index more useful. For example, it would be helpful if a user can create categories such as 'Definition' and 'Example' to organize query results, such as shown in Figure 1. Once these categories are created, the user can drag and drop hits into categories. The user can also create copies of a search result, and then place it in multiple categories. Finally, the current index state can be saved in JSON format, and then later loaded to generate the same tree structure again.[2]

Our approach employs Mel Frequency Cepstral Coefficients (Davis and Mermelstein, 1980) and a variation of Dynamic Time Warping algorithm called Segmental Dynamic Time Warping (Park and Glass, 2005). We have evaluated our system using videos from introductory Linear Algebra courses recorded at two different U.S. institutions (RIT and MIT). At RIT, a set of linear algebra lectures was recorded using a lone video camera in a classroom without students by one of the authors (Dr. A. Agarwal). Using queries recorded on a laptop by the instructor, our system achieved a Precision at 10 of 71.5%. Using the same queries extracted from the lecture audio, a Precision at 10 of 79.5% was obtained. The MIT lectures were recorded by an instructor with a different accent who used a lapel microphone for recording. Without modifying system parameters and using keywords similar to that used for the RIT lectures we obtained a much higher Precision at 10 of 89.5%, suggesting that our system is robust to different speakers and recording environments.

In the remaining of this paper, we summarize related work in Section 2, our keyword spotting methodology in Section 3, the experimental design and results in Sections 4 and 5, and then conclude and identify future directions in Section 6.

2. Related Work

Previous systems have been proposed for indexing, retrieving and annotating video content. For example, the MIT Lecture Browser by (Glass et al., 2007) allows users to search lecture audio using text queries. Automatic speech recognition is used to create a transcript of the lecture audio, which can then be searched textually. This transcription-based index may not have temporal information, and may contain recognition errors for rarer terms outside the language model. Similar to the MIT Lecture Browser, the Speech@FIT Lecture Browser (Szoke et al., 2010) uses speech recognition to support text search of lecture audio. This system shares many of the strengths and weaknesses of the MIT Lecture Browser. It also detects lecture slide changes using image features to provide pointers for lecture navigation.

The Video Audio Structure Text Multimedia (VAST MM) Browser designed by (Haubold and Kender, 2007) is another example of an indexing and annotation system designed for video presentations. This system creates a visual index for speaker segmentation using changes in activities. It also offers textual indices for searching through the transcription of the video.

NTU Virtual Instructor (Lee et al., 2014) offers sophisticated tools for finding lecture recordings of interest, including automatic summarization and keyword detection. Keywords are linked to particular points in the lecture in which they occur, allowing the user to rapidly find relevant content. Bilingual automatic speech recognition is integral to the approach, which also supports text-based search of spoken terms.

While these systems support lecture annotation and textual search, they do not offer video search using audio queries. In our work we seek to support audio queries, and avoid the need to train speech recognizers for new lecturers. To do this, we have chosen to use unsupervised keyword spotting in audio.

Mel Frequency Cepstral Coefficients (MFCC) are frequently used to represent speech audio in keyword-spotting systems. MFCC features were first discussed by Davis and Mermelstein (Davis and Mermelstein, 1980). MFCCs are computed based on a model of how human ears perceive speech, and compensate for insignificant variations present in higher frequency bands. MFCC feature extraction is usually followed by normalization to reduce the impact of environmental mismatch. (Alam et al., 2011) have discussed different normalization approaches for MFCC features. The short-term mean variance approach is similar to the whitening process used in this paper. However, they have used *mean* (μ) and *standard deviation* (σ) values

[1] The working demo of interface is available at `https://www.cs.rit.edu/~dprl/keywords/index.html`

[2] This prototype is created using 'jsTree' `http://www.jstree.com/`

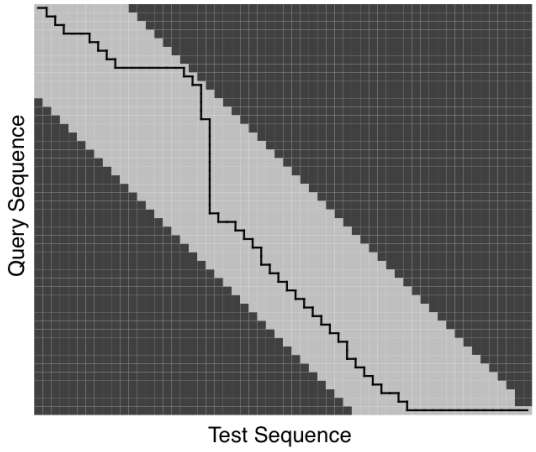

Fig. 2. Radius Constraint on SDTW Alignment. Frames must be aligned within the gray region, defined by a *radius* of ± 10 frames. This avoids aligning frames distant in time. The black line shows an alignment with longer horizontal and vertical segments representing skipped frames in the test and query sequences, respectively.

computed over a moving window instead of the complete sequence as done in this paper. Using parameters obtained from the complete sequence reduces processing time, which is important for a real-time system.

Noise reduction is used to remove non-speaker audio elements. (Doblinger, 1995; Kim and Stern, 2012) have discussed Cepstral Subtraction from the MFCC features for noise reduction. These techniques model slowly changing noise using a filtering approach. A noise profile is computed by filtering the input audio asymmetrically if the current intensity value of the MFCC feature is higher than the output of the filter from the previous frames. This resultant noise profile represents slowly changing audio signals which are considered to be noise. The noise profile is computed for each MFCC feature. Noise signals are then subtracted from the original feature values.

Dynamic Time Warping (DTW) is a common technique for computing the similarity of two temporal sequences. DTW is very similar to the *edit distance* algorithm. The edit distance algorithm calculates the distance between two strings where the cost of insertion, deletion and substitution is constant. Dynamic Time Warping also calculates the distance between two temporal sequences non-linearly, allowing 'warping' through the insertion and deletion of frames in each sequence, but cost is dependent upon the differences between matched feature vectors.

For keyword spotting, the query is usually much shorter than the test sequence. The standard DTW algorithm will stretch a query over the entire test sequence to evaluate similarity. Even if such matchings have low cost they are impractical, as a word utterance cannot realistically be spread across the test sequence.

To avoid this shortfall, (Park and Glass, 2005) have proposed an improved version of Dynamic Time Warping called Segmental Dynamic Time Warping (SDTW). SDTW uses a restriction on the warping path first proposed by (Sakoe and Chiba, 1978). In SDTW each frame of the query can be matched with a restricted number of frames of a test sequence based on a matching radius r. Due to this restriction, the warping of each query of length n is restricted within $(n-r)$ to $(n+r)$ frames of the test sequence (see Figure 2). Matching is then performed upon different segments of the test sequence, each starting at a different frame. The *step size* parameter controls the number of frames between start points for alignment on the test sequence.

3. Methodology

Our keyword spotting system is divided into three parts: Mel Frequency Cepstral Coefficients (MFCC) extraction, feature normalization, and Segmental Dynamic Time Warping to locate candidate matches.

3.1. MFCC Feature Extraction and Query Trimming

We have used the Sphinx-4 (Walker et al., 2004) library to extract MFCC features from speech recordings. While calculating these features, Sphinx considers 25ms frames, generated every 10ms. Hence, there is an overlap of 15ms between two consecutive frames. Each frame is then transformed to the frequency domain using Discrete Fourier Transform. The separated bands are then passed through triangular filters placed at logarithmic distances. The final features are calculated by transforming the resulting frames in the time domain using the Discrete Cosine Transform. The final feature vector contains 13 MFCC features. To reduce the impact of environmental noise, we then modify the MFCC features using the Sphinx 4 denoising module (Doblinger, 1995; Kim and Stern, 2012). We then add first and second order derivatives of the resulting MFCC features, producing a 39-element vector.

Queries may unintentionally have leading or succeeding pauses that are hard to detect by listening. These pauses can have a substantial negative impact upon retrieval performance, and so we have created a trimming function to remove them. As seen in Figure 3, the first raw MFCC feature in queries often has the largest values and variance. We observed empirically that the first MFCC feature associated with a pause has a value less than -10. To trim silence from queries, we remove consecutive frames in the forward and backward directions where the first raw MFCC value is less than -10.

3.2. Feature Normalization

While lectures are recorded in a classroom environment, queries to AccessMath may come from the lecture audio itself, or from a recording made by the instructor in a different environment (e.g. on a laptop in the instructor's office). Our normalization process is designed to reduce the impact of these changes in the recording environment.

Figure 3 shows raw MFCC feature values for a laptop-recorded query, where the first (lowest) MFCC has much larger values than those for higher frequencies due to the proximity of the microphone. When this query is used for retrieval, the first MFCC feature dominates the warping cost, mostly ignoring information in higher frequencies (e.g. 'thuds' are matched to keywords). Without normalization, retrieval precision is poor (see Section 5.1).

Normalization is divided into two steps. First, all 39 MFCC features are transformed using a whitening transform. Let μ_i

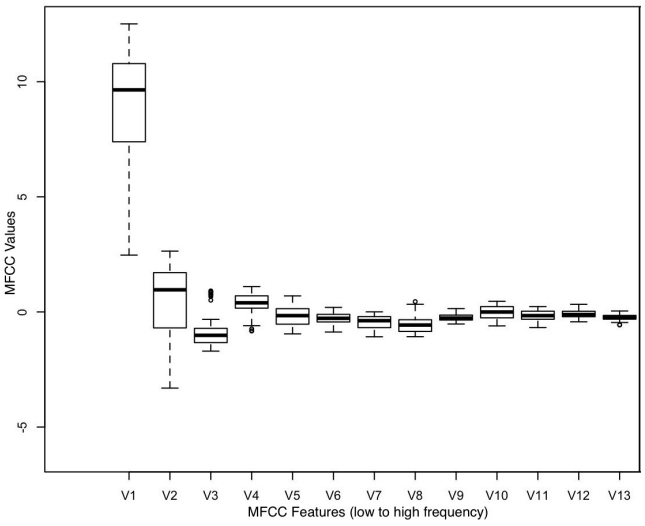

Fig. 3. Raw MFCC Value Distributions for Laptop-Recorded Query 'solution.' Intensities are much higher in the lowest frequency MFCC bands. As a result, low frequency MFCC values dominate warp costs for SDTW.

and σ_i represent the *mean* and the *standard deviation* for feature f_i in a sequence. Whitened value w_i for feature f_i is given by:

$$w_i = \frac{f_i - \mu_i}{\sigma_i} \tag{1}$$

This reduces the low-frequency domination of warping costs by producing features with similar distributions.

This whitening procedure is similar to short term mean variance normalization (Alam et al., 2011). As our datasets are relatively small, μ_i and σ_i values may be computed using all MFCC feature vectors in queries and lectures independently, without the need to use a small window for normalization. Each frame of speech input is whitened based on the value of μ and σ for that input (i.e. a query recording or lecture video).

In the second step, whitened MFCC feature vectors are converted to unit vectors. This ensures that all frames have equal weight when computing warp costs.

3.3. Segmental Dynamic Time Warping

Finally, to locate and score candidate matches of a query, we use Segmental Dynamic Time Warping (SDTW). Segmental Dynamic Time Warping attempts to align two temporal sequences non-linearly using dynamic programming. In our work we have used Euclidean distance as our distance metric. However, any other valid metric for computing the difference between two frames can be substituted.

Consider a query sequence Q and the sub-sequence S of the test sequence containing m and n frames respectively. The Dynamic Time Warping (DTW) algorithm will calculate the m by n matrix C (Müller, 2007) to determine a minimum cost warping path as shown in Figure 2. The cost of matching first frame S_0 of the test sequence against the first frame Q_0 of the query sequence is calculated using the Euclidean distance $d(Q_0, S_0)$. The remaining values of the first row and the first column of cost table C are calculated using Equation 2.

$$\begin{aligned} C_{0,j} &= d(Q_0, S_j) + C_{0,j-1} \\ C_{i,0} &= d(Q_i, S_0) + C_{i-1,0} \end{aligned} \tag{2}$$

After calculating the first row and first column values, the remaining values of the cost table can be calculated using the recursive formula expressed by Equation 3.

$$C_{i,j} = d(Q_i, S_j) + min \begin{cases} C_{i,j-1} \\ C_{i-1,j-1} \\ C_{i-1,j} \end{cases} \tag{3}$$

Finally the value of $C_{m,n}$ gives the minimal warping cost for the query against the test sequence. The warping path can be calculated backwards by tracing back the cost starting from the warping cost $C_{m,n}$.

As discussed earlier, this warping path will not be of practical importance if it is spread over a very long portion of the test sequence. To avoid impractical warping paths we have created a Java implementation for a modified version of the Segmental Dynamic Time Warping (SDTW) algorithm discussed in Section 2 (Zhang and Glass, 2009). Consider a short query sequence Q containing 150 frames being matched against a longer test sequence T containing 2000 frames. Using the radius constraint r, a given frame Q_i on the query can only be matched to frames within at most r frames away on the test sequence. Consider $r = 15$. The first frame of query Q_0 can be only matched to the frames between $(T_0 - T_{15})$. Similarly, the query frame Q_{15} can be matched within a range of $(T_0 - T_{30})$. Hence, any frame i can only match frames in the range $(i - r)$ to $(i + r)$. Due to this restriction, the first warping of the query starts at T_0 and ends between frames T_{135} to T_{165}. A restricted warping path is shown in Figure 2.

The next warping starts based on the value of the forward step parameter s. Consider $s = 20$. In this situation, the next warping will start at the frame T_{20} and ends between T_{155} and T_{185}. The degree of overlap between query warping paths depends upon the step parameter. Smaller step values result in better matching between query and test sequence, but also longer processing times.

SDTW avoids impractical warping paths without affecting the quality of matches, and reduces processing time for the algorithm. For each frame of the test sequence warping is restricted to frames within r frames. The time complexity for comparison is reduced from $O(mn)$, where m and n are sequence lengths, to $O(mr)$.

4. Experiments

Our goal in not requiring instructors to use a lapel microphone is to reduce the setup effort needed for capturing lectures, and to accommodate instructors who do not like working (directly) with microphones. We also want to allow lecturers to record their query keywords in a different environment than classrooms. We performed four experiments using two different datasets to measure the impact of change in recording environment and environmental mismatch between a query and a lecture on the performance of the system.

Evaluation Metric and Protocol. The experiments are evaluated using Precision at N. A Precision at 10 of 70% indicates that 7 of the top 10 hits for a query are valid (i.e. acoustically and semantically related to the query). We use the term average precision to represent the average Precision at k value over all queries, i.e. the average Precision at 10 is the average of the Precision at 10 scores for all queries.

To compute Precision at N values, hits were assigned by the first author to one of 5 categories: *1. Exact*, *2. Insertion* (extra sounds), *3. Delete* (missing sounds), *4. Both* (insertions and deletions) , and *5. No Match* (i.e. false positive). We use this categorization to divide similar but non-identical hits based on the addition and/or removal of utterances from the query.

In addition to the five acoustic categories for search hits, the *Insertion* and *Both* categories have two sub-categories used to identify whether a hit is semantically similar to the query. For example, consider the query 'dimensional,' and two top-10 search hits 'one dimension' and 'two dimensions.' Each hit is acoustically similar to the query with utterances both removed and added, and so these are assigned to the *Both* category, and the *Similar* sub-category. When a user searches with such a short query, results with similar sounds and meaning may be considered acceptable - due to the query brevity, it is highly likely that the user wants to find matches related to the stem of the query ('dimension'). Users seeking more specific utterances would likely have searched with a longer query such as 'high dimensional'.

Based on this reasoning, valid hits are defined as those belonging to the *Exact* match and *Insertion* categories (where the query term appears identically with sounds preceding and/or following the query), along with matches in the semantically *Similar* sub-categories for *Delete* and *Both* hits. Acoustically dissimilar hits in the *No Match* category and acoustically related but semantically dissimilar matches in *Delete* and *Both* are treated as misses (i.e. invalid hits).

MFCC Features and SDTW Parameters. For these experiments, we used the MFCC feature extraction described in the previous section, with a *radius* value of $r = 8$ frames, and testing alignment with the query keywords starting from every MFCC frame in a lecture (i.e. using a *step size* of $s = 1$). The small step size leads to more accurate keyword detection, but also longer processing times.

Computational Resources. We used a server with 96GB of RAM and an Intel Xenon CPU with 24 processors, each with a clockspeed of 2.93GHz for all experiments. Queries were executed in parallel over available videos, after which results were pooled and ranked.

4.1. RIT Linear Algebra Lectures

We want to provide the facility of recording queries outside the classroom in a different environment. We also expect users to be interested in re-querying the system using a result generated by a laptop-recorded query, and to extract queries directly from lecture audio. We performed three experiments using RIT Lecture Dataset to measure the performance of the system using these different query sources.

Dataset. 18 lectures for an introductory Linear Algebra course of between 45 and 60 minutes each was given by an RIT Math Professor (Dr. A. Agarwal, one of the authors). The instructor speaks with an Indian accent. The lectures were recorded using a single video camera in a small classroom environment without students, but with two assistants in the classroom typing notes. These lectures contain some noise, including typing noise, and the opening and closing of doors. We selected the left audio channel, to work with a monaural signal. The total duration of the lectures is around 1000 minutes.

Queries. After lectures had been recorded, our RIT Math Professor created a list of twenty keywords deemed useful for indexing his videos. We have used three sets of recordings for query keywords with the RIT dataset. This list of keywords can be seen in Table 2.

- **Experiment 1 (laptop).** Query keywords are recorded on a laptop by the instructor. The longest query and the shortest queries have lengths of 1.62s and 0.45s for this set. The mean length of a query is 0.74s with a standard deviation of 0.28s.
- **Experiment 2 (re-query).** Search hits generated for *Experiment 1* are used to query the system. Hits are selected from the top 10 results generated for *Experiment 1*. These are not necessarily the first result generated by the system, but rather the one deemed to be most phonetically similar to the query. The longest query and the shortest queries have lengths of 1.31s and 0.29s for this set. The mean length of a query is 0.76s with a standard deviation of 0.21s.
- **Experiment 3 (manual).** Queries are manually extracted from lecture recordings. The longest query and the shortest queries have lengths of 2s and 0.37s for this set. The mean length of a query is 0.73s with a standard deviation of 0.35s.

4.2. MIT Linear Algebra Lectures

As MFCC features are sensitive to both the speaker and environment (Aradilla and Bourlard, 2008), we carried out an experiment to test generalization of our system to new speakers and environments, i.e. the robustness of the system. Aside from the use of a new set of Linear Algebra lectures by another instructor, the experimental set-up is identical to that for the first three experiments.

Dataset. We used a set of introductory Linear Algebra lecture videos by Dr. Gilbert Strang available from MIT OpenCourseWare.[3] Dr. Strang speaks with an American accent. The dataset contains 35 lectures of 40 to 50 minutes each. The lectures are recorded in a live classroom with students, with the instructor wearing a lapel microphone. The use of the lapel microphone reduces the volume of environmental noise relative to the speaker's voice. There are approximately 1300 minutes of video for the course, roughly 300 minutes longer than for the RIT lectures. Unlike for the RIT lectures, a complete transcript for all lectures was available.

Queries. Similar to Experiment 3, queries were extracted manually from the MIT lecture videos. We used 15 common

[3] 18.06 Linear Algebra, Spring 2010. `http://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010`

Table 1. Categorization of Top-10 Results for Three Laptop-Recorded Queries. Hits missing part of the query (*Delete*, *Both*) are further categorized as semantically *Similar* or *Dissimilar.*

	Acoustic Match with Query						
	Exact	Insert	Delete		Both		No Match
			Similar	*Dissimilar*	*Similar*	*Dissimilar*	
dimensional	-	2	-	-	7	-	1
multiplicative	1	-	-	-	3	-	6
solution	-	10	-	-	-	-	-

keywords for evaluating the performance of the system on both datasets which are shown in Table 2. Three keywords out of these 15, shown in parenthesis, are slight modification of original keywords as the original keywords were completely absent or the modified form was highly preferred in MIT lectures. These keywords represent the same concept with a slightly different word form. However, lecture transcripts showed that five keywords were completely absent in MIT lecture. The RIT Math Professor (our co-author) suggested additional terms conceptually related to the missing terms. As a result, for the MIT lectures the query keywords *homogeneous, non-homogeneous, consistent, linear span* and *closure* were replaced by *pivot, elimination, linear combination, invertible* and *identity matrix*.

- **Experiment 4 (manual)** The experimental setup is similar to the first three experiments, differing only in the query and lecture recordings used. The query set contains twelve identical, three modified and five distinct keywords/keyphrases relative to the previous experiments.

5. Results

5.1. RIT Lectures (Experiments 1-3)

In **Experiment 1 (laptop)** the average Precision at 10 over the 20 laptop-recorded queries was 71.5%, with a standard deviation of 29.6%. While scores for individual queries in the top-20 are omitted for reasons of space, the average Precision at 20 was 63% for this experiment. As discussed in the previous Section and illustrated in Table 1, the number of valid hits is the number of hits assigned to the *Exact* and *Insert* categories, along with semantically *Similar* hits where sounds are missing (*Delete*) or where sounds are both added to and missing from the query (*Both*).

A visualization of hit categorization for three queries is shown in Table 1. As the SDTW algorithm considers the warping path between the length of the query within a given radius, and that pronunciation in lecture was generally faster, the path with lowest warping cost usually leads to generated hits preceded and/or followed by additional sounds, as seen by the concentration of hits in the *Insert* and *Both* categories. For many of the laptop queries, results were concentrated within the *Exact* and *Insert* categories. For queries like 'multiplicative' and 'dimensional,' hits are concentrated in the *Both* hit category. This is expected as root forms 'multiply' and 'dimension' are very common in linear algebra. Many hits contain both insertion and deletion of sound for these queries.

Queries 'row-reduced', 'system of equations', 'zero-vector' and 'echelon form of a matrix' have 100% Precision at 10. These four queries have a distinct pronunciation, and the term

Table 2. Valid Top-10 Hits for Experiments 1-4

	RIT		RIT	MIT
	LAPTOP	REQUERY	MANUAL	
Query	EXP 1	EXP 2	EXP 3	EXP 4
augmented	5	**10**	**10**	3
dimensional	9	8	**10**	**10**
row-reduced	**10**	**10**	**10**	
(reduced row)				9
solution	**10**	**10**	**10**	**10**
system of equations	**10**	**10**	**10**	8
transpose	9	**10**	**10**	9
zero vector	**10**	**10**	**10**	9
echelon form of a matrix	**10**	**10**	9	
(echelon form)				**10**
independent	8	4	9	**10**
multiplicative	4	3	8	
(multiplication)				**10**
system	**9**	8	7	**10**
variables	6	4	6	**10**
coefficients	8	**9**	5	8
reduce	3	**10**	5	4
orthogonal	1	3	4	**10**
RIT ONLY				
homogeneous	7	**10**	**10**	
linear span	6	**10**	9	
consistent	**9**	**9**	**9**	
closure	**8**	6	7	
non-homogeneous	1	**2**	1	
MIT ONLY				
elimination				**10**
linear combination				**10**
identity matrix				**10**
pivot				**10**
invertible				9
MEAN (μ)	7.15	7.80	7.95	**8.95**
STDEV (σ)	2.96	2.93	2.56	**1.97**

is not connected with other syllables in the lecture recordings. However, such strong results were not observed with the query 'non-homogeneous'. The pronunciation of prefix 'non' is usually blended with 'homogeneous' in the lecture videos. For the laptop-recorded query, the pronunciation of 'non' and 'homogeneous' is detached. These hits are categorized in *Both*, with sub-category *Dissimilar*, as they are phonetically but not semantically similar to the query. For example, even though the pronunciation of 'are homogeneous' resembles 'non-homogeneous', the concepts associated with these are directly opposed to one another.

Another example of acoustically similar but semantically distinct hits was seen for 'reduce.' Many acoustically similar top-10 hits are obtained, such as 'introduce' and 'produce,' due to a much longer time spent pronouncing 'duce' than 're,' and so strong matches to the suffix of the query are likely weighted higher. However, these hits are not semantically related to the query, and are treated as misses (false positives).

Effect of Whitening. In an earlier preliminary experiment we found whitening greatly improves performance - without whitening, for the same set of 20 laptop-recored queries, average Precision at 10 was less than half, at 32%.

Accelerating Laptop Queries. Given that our laptop queries

were often slower than utterances in-lecture, we investigated increasing the speed for a subset of the laptop queries. The query 'variable' obtained a precision at 10 of 60% for the original laptop recording of the query. This increases to 80% when the query was increased to 1.05 times the original speed. A further increase in speed resulted in a decrease in precision, due to increasing distortion of the audio. A similar pattern was observed for the query 'orthogonal'. The performance for all queries in our subset decreased with increases in speed beyond 1.10 times the original duration. The pronunciation of the query becomes different beyond this speed, which affects the extraction of MFCC features.

For **Experiment 2 (re-query)**, we have used the top hit generated by Experiment 1 as a query. The average Precision at 10 is 78% with a standard deviation of 29.3%. The average Precision at 10 is higher than for the laptop recorded queries. In this experiment, ten keywords have perfect Precision at 10 (100%). For many queries, the top hit is assigned to the *Insert* category. For the query 'variables,' we used the closest hit, which was 'pivot variable'. As a result, hits which do not contain 'variable' are produced due to their similarity with 'pivot'. A similar phenomenon was observed for queries 'solution' and 'transpose'. These results also expanded the set of results obtained by the original laptop queries in a manner similar to that described below for Experiment 3.

For **Experiment 3 (manual)**, an average Precision at 10 of 79.5% with a standard deviation of 25.6% was obtained for queries manually extracted from the lecture recordings. This is to be expected, as queries extracted from lectures are from the same acoustic environment, and have 'within-lecture' pronunciations (in particular, the lecturer speaks more quickly and with differing emphasis in the lecture recordings than in the laptop recordings).

Queries extracted from lectures generated numerous top-10 hits not found in the top-10 for laptop recorded queries. The results generated for laptop-recorded queries and extracted queries have an overlap of 19.5% in the top-10 results. For example, for 'augmented', three results generated by the laptop-recorded query were captured by the query extracted from the lectures. The within-lecture query captured seven new hits from the video lectures, and all hits were valid (Precision at 10 is 100%).

5.2. MIT Lectures (Experiment 4)

In **Experiment 4**, the system achieved average precision at 10 of 89.5%. This exceeds the results for *Experiment 3*, which also contains queries extracted from lectures by 10%. Except for queries 'augmented' and 'reduce' all the queries have very strong results. For the 15 common queries, *Experiment 3* has precision at 10 of 82% while *Experiment 4* has precision at 10 of 86.67%, which are comparable results. The overall gain in the performance can be attributed to lecture recordings using a lapel microphone, which picks up less environmental noise.

Dr. Strang speaks emphatically in the MIT lectures, which divides the query 'augmented' in two strong segments, 'aug' and 'mented'. Due to this reason, 'augmented' is often confused with some sound divided into two strong parts separated by a small pause. One the other hand, 'reduce' occurs only twice as a separate word. Mostly it is present in some other word form as 'reduced' or 'reduction'. Fewer occurrences coupled with its similarity with other words, such as 'produce', results in lower precision.

Based on the results obtained with MIT dataset, it appears that our system generalizes reasonably well to new recordings. Our system is also robust enough to produce strong results for both lapel and camera microphone recordings.

5.3. Retrieval Times

The average run time for RIT lectures dataset was 108 seconds with a standard deviation of 60 seconds. The longest running query was 'echelon form of a matrix' which is expected as it is the longest query. The query 'system' took the shortest time to run, at 67 seconds. The MIT lectures dataset had a similar average running time of 97 seconds, with a standard deviation of 40 seconds. The longest running query was 'system of equations' (189 seconds) and 'pivot' was the shortest (38 seconds).

Even though MIT lectures dataset is longer than RIT lecture dataset their run time are comparable. In RIT lecture dataset, 18 lectures are divided in total of 56 recordings while MIT dataset contains only 35 recordings, one for each lecture. Our system performs whitening and normalization based on the audio recording in a single file. As a result, the run time are comparable as the MIT videos are longer, but the RIT dataset contains more videos.

5.4. Comparison with State-of-the-Art

The performance of our system is comparable to state-of-the-art keyword spotting systems. (Zhang and Glass, 2009) used Segmental Dynamic Time Warping algorithms with Gaussian posteriorgrams to create a keyword spotting system. They obtained an average Precision at 10 of 68.3% for an MIT lecture dataset (Glass et al., 2004), comparing test queries against segmented utterances. The dataset from which the utterances were segmented contains nearly 300 hours of audio from different classes at MIT. This data is collected in a classroom environment with a lapel microphone (e.g. as found in the MIT recordings used for Experiment 4). (Aoyama et al., 2014) used the same dataset for keyword spotting using graph-based search. They obtained an average Precision at 10 of roughly 80% for 20 keywords compared against 3000 test utterances. In contrast, the AccessMath keyword search system identifies keywords in complete lecture recordings, without a prior segmentation step.

Both of these systems require training to create Gaussian Posteriorgrams or a graph index. Training GMMs requires segmenting files into speech and non-speech segments, and selecting the number of component Gaussians to include in the mixture model. Our approach does not require prior segmentation of audio files, and requires no training aside from setting the query trimming threshold, SDTW radius, and SDTW step size parameters. The parameter values used in this paper seem reasonably robust, given that for the RIT and MIT lectures strong results were obtained for two different lecturers with different

accents, and in two different classroom environments with different recording setups (with average Precision at 10 values of 79.5% and 89.5% for within-lecture queries, respectively).

A limitation of our approach is that it is designed for use with single rather than multiple speakers, while (Aoyama et al., 2014; Zhang and Glass, 2009) support multiple speakers. However, for our intended application, single-speaker detection is sufficient.

6. Conclusion

We have proposed a within-speaker keyword spotting system to assist instructors with indexing their lecture videos, in order to help low-vision students to more easily locate topics in the videos. Our system has an average Precision at 10 of 71.5% for laptop recorded queries for Linear Algebra lectures recorded at RIT, and 79.5% and 89.5% for RIT and MIT lectures respectively using within-lecture queries. We have also created a prototype for improving the accessibility of generated hits and organizing them into a tree structured index. Our keyword spotting system is unsupervised, and generates results using Segmental Dynamic Time Warping without any training data. It is also robust enough to compensate for changes in speaker, recording environment and recording setup.

In the future, it is possible to improve the accessibility of this system by including the within-lecture searching functionality, and support for re-querying using search hits. Sometimes an instructor pronounces a word in a peculiar way, e.g. slow, fast or stretched over a long interval. It may be useful to expand the system to include query expansion techniques, such as by changing the speed of queries.

We are able to reduce run times for indexing significantly with parallelization. However, the system still requires noticeable time to create an index. It would be interesting to explore other techniques such as Information Retrieval based DTW (Anguera, 2013), Randomized Acoustic Indexing and Logarithmic Time Search (Levin et al., 2015) or pre-computing similarity matrices using HMM (Chung et al., 2014). These methods claim to improve speed and memory footprint over traditional DTW without affecting the accuracy significantly.

Apart from these points, we would also like to improve accessibility by providing a within-lecture query facility in the system interface, and to include a facility of creating an index from the web interface itself. Our system has been designed primarily to support linear algebra lectures. However, we would like to explore the performance and adaptability of the system in different domains where lectures consist of classroom discussion between students and the lecturer.

Acknowledgments

We thank the anonymous reviewers for suggestions that greatly improved this paper, and Ben Miller-Jacobson for assistance with revising the paper. This work was supported by the National Science Foundation (USA) under Grant No. HCC-1218801.

References

Alam, M., Ouellet, P., Kenny, P., O'Shaughnessy, D., 2011. Comparative evaluation of feature normalization techniques for speaker verification, in: Advances in Nonlinear Speech Processing. Springer Berlin Heidelberg. volume 7015 of *Lecture Notes in Computer Science*, pp. 246–253.

Anguera, X., 2013. Information retrieval-based dynamic time warping., in: INTERSPEECH, pp. 1–5.

Aoyama, K., Ogawa, A., Hattori, T., Hori, T., Nakamura, A., 2014. Zero-resource spoken term detection using hierarchical graph-based similarity search, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7093–7097.

Aradilla, G., Bourlard, H., 2008. Posterior-based features and distances in template matching for speech recognition, in: Proceedings of the 4th International Conference on Machine Learning for Multimodal Interaction, Springer-Verlag, Berlin, Heidelberg. pp. 204–214.

Chung, C., Chan, C., Lee, L., 2014. Unsupervised spoken term detection with spoken queries by multi-level acoustic patterns with varying model granularity, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014, pp. 7814–7818.

Davis, S., Mermelstein, P., 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on, Acoustics, Speech and Signal Processing 28, 357–366.

Doblinger, G., 1995. Computationally efficient speech enhancement by spectral minima tracking in subbands, in: Proc. Eurospeech, pp. 1513–1516.

Glass, J., Hazen, T., Cyphers, S., Malioutov, I., Huynh, D., Barzilay, R., 2007. Recent progress in the mit spoken lecture processing project, in: Proceeding of the 8th Annual Conference of the International Communication Association, pp. 2553–2556.

Glass, J., Hazen, T.J., Hetherington, L., Wang, C., 2004. Analysis and processing of lecture audio data: Preliminary investigations, in: Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004, Association for Computational Linguistics, Stroudsburg, PA, USA. pp. 9–12.

Haubold, A., Kender, J.R., 2007. Vast mm: Multimedia browser for presentation video, in: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, ACM, New York, NY, USA. pp. 41–48.

Kim, C., Stern, R.M., 2012. Power-normalized cepstral coefficients (pncc) for robust speech recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 4101–4104.

Lee, H.Y., Shiang, S.R., Yeh, C.F., Chen, Y.N., Huang, Y., Kong, S.Y., Lee, L.S., 2014. Spoken knowledge organization by semantic structuring and a prototype course lecture system for personalized learning. Audio, Speech, and Language Processing, IEEE/ACM Transactions on 22, 883–898.

Levin, K., Jansen, A., Durme, B.V., 2015. Segmental acoustic indexing for zero resource keyword search, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015, pp. 5828–5832.

Müller, M., 2007. Information Retrieval for Music and Motion. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Park, A., Glass, J., 2005. Towards unsupervised pattern discovery in speech, in: IEEE Workshop on, Automatic Speech Recognition and Understanding, 2005, pp. 53–58.

Rabiner, L.R., Rosenberg, A.E., Levinson, S.E., 1978. Considerations in dynamic time warping algorithms for discrete word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 26, 575–582.

Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on, Acoustics, Speech and Signal Processing 26, 43–49.

Szoke, I., Cernocky, J., Fapso, M., Zizka, J., 2010. Speech@fit lecture browser, in: Spoken Language Technology Workshop (SLT), 2010 IEEE, pp. 169–170.

Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., Wolf, P., Woelfel, J., 2004. Sphinx-4: A flexible open source framework for speech recognition. Technical Report.

Zhang, Y., Glass, J.R., 2009. Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams, in: IEEE Workshop on Automatic Speech Recognition & Understanding, 2009. ASRU 2009., IEEE. pp. 398–403.