Spoken Knowledge Organization by Semantic Structuring and a Prototype Course Lecture System for Personalized Learning

Hung-yi Lee, Sz-Rung Shiang, Ching-feng Yeh, Yun-Nung Chen, Yu Huang, Sheng-Yi Kong and Lin-shan Lee *Fellow, IEEE*

***Abstract*—It takes very long time to go through a complete online course. Without proper background, it is also difficult to understand retrieved spoken paragraphs. This paper therefore presents a new approach of spoken knowledge organization for course lectures for efficient personalized learning. Automatically extracted key terms are taken as the fundamental elements of the semantics of the course. Key term graph constructed by connecting related key terms forms the backbone of the global semantic structure. Audio/video signals are divided into multi-layer temporal structure including paragraphs, sections and chapters, each of which includes a summary as the local semantic structure. The interconnection between semantic structure and temporal structure together with spoken term detection jointly offer to the learners efficient ways to navigate across the course knowledge with personalized learning paths considering their personal interests, available time and background knowledge. A preliminary prototype system has also been successfully developed.**

***Index Terms*—course lectures, spoken content retrieval, speech summarization, keyterm extraction**

I. Introduction

The necessity of life-long learning in the era of knowledge explosion together with the ever-increasing bandwidth of Internet and continuously falling costs for memory bring about the rapid proliferation of Massive Open Online Courses (MOOCs). Instructors post slides and video/audio recording of their lectures on on-line lecture platforms, and learners can easily access the curricula. The worldwide online learners working in different technical areas with different background knowledge have widely varying learning requirements. For example, the novices of a subject may need an effective way to comprehend the high-level core concepts in the subject, while some experts may need an easy way to review the low-level details of a specific subtopic of the subject. As a result, new techniques for personalized learning helping all different learners properly utilize the curricula in their own most efficient way and plan their own personalized learning paths are highly desired but still missing for on-line lecture platforms today.

A major difficulty for the many different learners to efficiently utilize the many complete course lectures available over the Internet is that it may not be easy for people in the busy world to spend very long time to go through a complete course (e.g. it may include tens of hours). With recent advances of spoken content retrieval [1], [2], it is now possible to search over the on-line lectures for some specific topics based on the audio information [3]–[6]. Good examples include MIT lecture browser [3] and Speech@FIT lecture browser [4]. Such lecture browsers enable the user to type a text query and receive a list of spoken segments within the lectures containing the query terms.

Direct retrieval over the course content for some specific topics may not always be helpful to the learners. The course content is usually semantically structured with one concept following the other. Without the background, it is often difficult to understand a retrieved paragraph of a course. Without the semantic structure of the content, it is difficult for the learners to come up with suitable queries to search for the target topics. Displaying the key terms extracted from lecture courses is an effective way to present to the learners the core concepts in the courses. FAU Video Lecture Browser displaying automatically extracted key terms to help the interactive assessment of video lectures [5] is a good example of such approaches.

Some on-line lecture platforms can summarize the audio/video recordings of the course lectures into compact versions. A good example is the lecture browsing system of Toyohashi University of Technology [7], [8]. With the summaries of the lecture recordings, novices can listen to the summaries for obtaining the core concept of the courses and selecting the right parts best fitting their needs before going through the complete version, and the students can also review the content of the courses very quickly.

In this paper, in order to help individual learners develop their personalized learning paths from an on-line lecture platform considering specific learning requirements, we present a new approach of spoken knowledge organization for the course lectures. We automatically extract key terms from the lectures and take them as the fundamental elements of the semantics for the spoken knowledge covered by the course content. We automatically connect related key terms to construct the key term graph as the backbone of the global semantic structure of the course. We divide the audio/video signals into paragraphs, sections and chapters as a multi-layer temporal structure of the course, and develop summaries for each paragraph, section and chapter as the local semantic structure. The global semantic structure of key term graph is then interconnected with the

This work was supported by the National Science Council of Taiwan under contract numbers NSC 100-2221-E-002-229-MY3.

nodes of the multi-layer temporal structure. All these are jointly referred to as semantic structuring here in this paper. The whole content of the course is then indexed by spoken content retrieval technologies, so the learners can efficiently navigate over the knowledge covered by the course with globally and locally structured semantics. This offers multiple ways to the learner to interact with the system and access the curricula in their personalized way. A preliminary prototype system was successfully developed at National Taiwan University (NTU), referred to as NTU Virtual Instructor. The first version was completed in 2009 [9], while this paper presents the latest version and the technologies used [10].

The rest of this paper is structured as follows. The proposed approaches are overviewed in Section II. The course corpus used in this research and the bilingual ASR techniques used for transcribing the spoken content are briefly summarized in Sections III and IV. The semantic analysis, key term extraction, key term graph construction, speech summarization and spoken content retrieval are then respectively presented in detail in Sections V, VI, VII, VIII and IX. The prototype system is described in Section X, and Section XI finally gives the concluding remarks.

II. Overview of the Proposed Approach

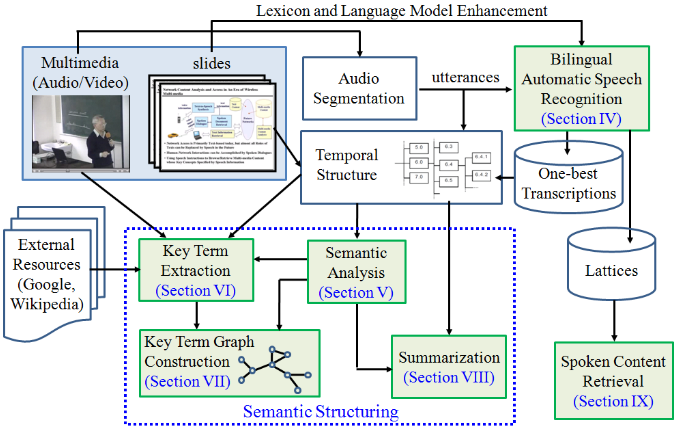

Lexicon and Language Model Enhancement
Multimedia (Audio/Video)
slides
Audio Segmentation
utterances
Bilingual Automatic Speech Recognition (Section IV)
Temporal Structure
One-best Transcriptions
External Resources (Google, Wikipedia)
Key Term Extraction (Section VI)
Semantic Analysis (Section V)
Lattices
Key Term Graph Construction (Section VII)
Summarization (Section VIII)
Spoken Content Retrieval (Section IX)
Semantic Structuring

Fig. 1: *Overview of the proposed approach.*

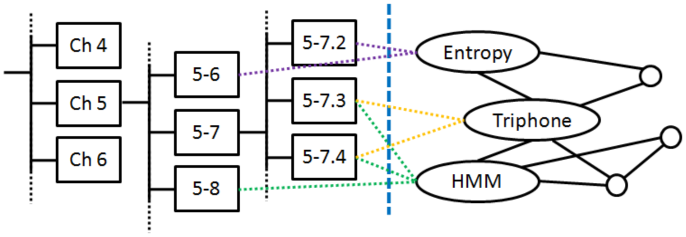


Fig. 2: *Interconnection between the semantic structure and the multi-layer temporal structure.*

An overview of the proposed approach is shown in Fig. 1. The course materials including slides (we assume slides for the lectures are available) and multimedia (synchronized audio/video) is at the upper left corner of Fig. 1. The audio signals (and therefore video signals) are first divided into utterance-level segments as in the top middle of the figure.

A. Automatic Speech Recognition (ASR)

An ASR system transcribes the utterance-level segments into lattices or one-best transcriptions at the upper right corner of Fig. 1. Correctly transcribing the spoken lectures is challenging [11]–[13], not only because the lectures are spontaneous, but because spoken lectures usually contain many technical terms or OOV words, so the texts of the slides are very helpful in enhancing the lexicon and language model used in ASR [8], [14], [15].

On the other hand, many lecturers with non-English native languages give the lectures primarily in their native languages (referred to as the host language here, such as Mandarin), but with some special terms produced in English (referred to as guest language) embedded within the utterances of the host language. This is because very often almost all special terminologies for the courses are directly produced by the lecturers in English without translating them into the host languages. Although only a small portion of signals in this corpus belongs to English, since most of them are terminologies, they should not be ignored. Since the dataset is very biased to Mandarin, special ASR techniques should be designed to transcribe this kind of corpus. Because such situation is very common for lectures offered in countries with non-English native languages, special efforts have been made and reported here to handle this problem. The ASR for such bilingual lectures is summarized in Section IV.

B. Multi-layer Temporal Structure

At the upper middle of Fig. 1, the temporal structure of the course is constructed in button-up three layers: paragraph, section and chapter. The paragraphs are groups of neighbouring utterance-level segments with similar lexical distributions in their transcriptions clustered with dynamic programming [16]. A paragraph is usually a part of a slide. The audio corresponding to a slide or a few slides with the same title is regarded as a section, usually containing several paragraphs. Since there exist software tools to synchronize the slides and the video/audio during recording, sometimes the slides and the video/audio recording are automatically synchronized, and the sections can be obtained directly. Otherwise, the video/audio recording can be aligned with the slides by hidden Markov modeling in which each slide is a state based on its text and the ASR transcriptions of a paragraph is an observation [16]. A chapter corresponds to a set of consecutive slides on a common subject, usually shown on the slides, which are defined by the instructor or a textbook. The paragraphs, sections and chapters are actually nodes on different layers in the multi-layer temporal structure.

C. Semantic Analysis

This is at the lower middle of Fig. 1 to be further described in Section V. Semantic analysis generates the latent topics of the course as well as some useful semantic parameters helpful to key term extraction, key term graph construction, and summarization as presented below.

D. Key Term Extraction

This is in the middle left of Fig. 1. Key terms are used here as the fundamental elements of the course semantics. They are automatically extracted based on not only the latent topic information but also audio information such as prosody and external resources such as Google and Wikipedia, as will be presented in Section VI. The extracted key terms are displayed in each node in the temporal structure (paragraph, section and chapter), so the learners can realize the core concepts discussed in a node by a glance at the key terms. The system can also show all the nodes (paragraphs, sections and chapters) containing a specific key term, so the learner can know how the key term is related to other parts of the course or learn the concept about the key term sequentially following the order it was discussed in the lectures.

E. Key Term Graph Construction

This is at the lower left corner of Fig. 1. The key term graph has all key terms extracted from the entire course as its nodes, with only those with high enough relationships linked by edges. This graph represents the backbone of the global semantic structure of the course. Each key term is connected to the paragraphs, sections and chapters (nodes in the temporal structure) in which the keyterms is included. Therefore, the semantic structure of key terms and the temporal structure are inter-connected through the key terms as shown in Fig. 2. In this way, the learner can easily find out related parts of the course which can be studied jointly. The details for keyterm graph construction will be reported in Section VII.

F. Speech Summarization

The summaries of the course content of the nodes in the temporal structure are generated as in the middle bottom of Fig. 1 (paragraphs, sections and chapters). Therefore, instead of listening to the whole audio/video recording (e.g. several hours for a chapter), a learner can skim the much shorter summaries and then decide if he wishes to go through the entire content in detail. This will be further discussed in Section VIII. The semantic analysis, key term extraction, key term graph construction and speech summarization mentioned above in Section II-C to II-F are jointly referred to as semantic structuring here in this paper.

G. Spoken Content Retrieval

The key term graph, summaries, semantic structure and temporal structure jointly build the course content in a structured way. However, the learner needs to be able to retrieve the spoken segments mentioning the concepts he wishes to learn. The lattices generated by ASR from the audio of the courses are indexed and retrieved for this purpose as on the right part of Fig. 1. When the learner enters a text query, the spoken content retrieval engine searches through the lattices and returns the utterance-level segments regarded as containing the query, together with the links to the paragraphs, sections or chapters it belongs to, since it makes better sense to listen to the complete paragraph, section or chapter for learning the concepts with the query. The details will be introduced in Section IX.

III. Corpus Description

The corpus used in this research was the lectures for a course of 45.2 hours long on Digital Speech Processing offered in National Taiwan University in 2006. There was a total of 17 chapters with 196 sections in the multi-layer temporal structure in Subsection II-B. The course slides were available, completely in English. The audio was recorded by the hand-held microphone with 16KHz sampling rate. The utterances in the lectures were produced spontaneously with many disfluencies such as pauses, hesitations, repairs and repetitions making the recognition more challenging. The intra-sentential code-switching is an extra problem. The instructor produced the whole lectures in the host language of Mandarin (the native language), but many words or phrases (primarily terminologies of the course) were naturally produced in English (the guest language) and embedded in the Mandarin utterances. For example, in the sentence, "除了 speech recognition 的技術之外，我們還需要 indexing 跟 retrieval 的技術 (Except for speech recognition technology, we also need technologies for indexing and retrieval.)", the phrase "speech recognition" and the words "indexing" and "retrieval" were produced in English, while other parts of the sentence were in Mandarin.

The whole corpus was divided into several partitions, and their detailed information is in Table I. The acoustic models used to transcribe the target lecture corpus were trained in two scenarios, speaker-dependent and speaker-adaptive. In speaker-dependent scenario, 9.1 hours speech in the corpus (*Training Set* in Table I) was used to train speaker dependent acoustic models; while in speaker-adaptive scenario, speaker independent acoustic models were adapted by 0.5 hour speech in the corpus (*Adaptation Set* in Table I). All parameters in acoustic model training procedures were tuned on a 2.1 hours development set (*Development Set* in Table I). The audio besides *Training Set*, *Adaptation Set* and *Development Set* in the corpus is referred to as *Testing Set*, which has the length of 33.5 hours. The recognition accuracies were evaluated on 2.2 hours of speech from *Testing Set*, and 40 sections (around 11.7 hours speech) in *Testing Set* were used for testing the performance of summarization. The key term extraction, key term graph construction and spoken content retrieval were all tested based on the complete *Testing Set*. Columns (b) and (c) in Table I are the numbers of Chinese characters[1] and English words respectively, and the numbers in the parentheses in columns (b) and (c) are the numbers of *unique* tokens. In *Testing Set*, there are respectively 93% and 71% of unique Chinese characters and English words included in *Training Set*. The ratios of English words to Chinese characters are in column (d). Column (e) shows the percentage of code-switched utterances in each set. From columns (d) and (e), we found that although only a small portion of the signals belongs to English words in the corpus, more than half of the utterances has code-switched phenomenon.

[1] For Mandarin Chinese, the positions of word boundaries are not uniquely defined, so the number of Chinese words in a sentence is not unique. Hence, we report the numbers of Chinese characters instead of words here.

	(a) Length (in hours)	(b) Number of Chinese Characters	(c) Number of English Words	(d) Ratio of English Words to Chinese Characters	(e) Percentage of Code-switched Utterances
Training Set	9.1	124K (1.2K)	10K (0.9K)	8.0%	53%
Adaptation Set	0.5	6.7K (0.5K)	0.6K (0.2K)	8.7%	54%
Development Set	2.1	30K (0.8K)	2.7K (0.5K)	8.8%	55%
Testing Set	33.5	305K (0.7K)	27K (0.5K)	8.8%	57%

TABLE I: Detailed information for the partitions of the target corpus. The numbers in the parentheses in columns (b) and (c) are the numbers of *unique* tokens.

We recruited graduate students of National Taiwan University who had taken the target course to annotate key terms, key term graph and reference summaries. There were 61 subjects annotating key terms. Since different subjects annotated quite different sets of key terms with different numbers, we assigned a score proportional to $1/N_j$ to a term if it was annotated by a subject j who selected a total of N_j key terms. In this way when a subject annotated less key terms, each of these annotated key terms received a higher score. We then sorted the terms by their total scores assigned by the 61 subjects, and selected the top $\bar{N}$ of them as the reference key terms, where $\bar{N}$ was the integer closest to the average of N_j for all subjects. A total of 154 key terms[2] (including 59 key phrases and 95 keywords) were generated as the reference key terms in this way. Examples of such reference key terms included "language model", "speech recognition", "name entity" (key phrases), "LVCSR", "n-gram" and "entropy" (keywords). Only 3 out of 59 key phrases and 4 out of 95 keywords were in Chinese. This shows that most terminologies carrying key information for the course were in English. Given this reference key term list, the 61 annotators achieved average precision, recall and F-measure of 66.13%, 90.30% and 76.37%. Based on these reference key terms, 12 subjects generated their key term graphs by connecting the key terms considered as relevant. To form one reference key term graph from the key term graphs generated by different subjects, we assigned a score proportional to $1/N_k$ to a pair of key terms if they were connected on a key term graph with N_k edges produced by subject k. Then the reference key term graph was generated by connecting the $\bar{N}'$ of key term pairs with the highest scores, where $\bar{N}'$ was the integer closest to the average of N_k for all subjects. The average Graph Edit Distance (GED) [17] from the reference key term graph to the key term graphs generated by the annotators was 0.066. Reference summaries for the 40 sections in *Testing Set* were generated by 15 annotators. The reference summaries were utterance-level segments selected from the sections. Each sections has 3 short and 3 long reference summaries generated by different annotators. For the short version, the length (number of Chinese characters plus English words in the manual transcriptions) of the summaries does not exceed 10% of the whole sections; for the long version, the length does not exceed 30%. The average kappa score between the annotators were 75% and 48% respectively for short and long summaries. This shows that the annotators agreed with each other more when generating the short summaries. For spoken content retrieval, 162 Chinese queries were manually selected as testing queries, each consisting of a single word.

IV. Recognizing Bilingual Code-switched Lectures

Transcribing bilingual corpus is difficult, because each acoustic event may belong to either language, and may form some words in either language when combined with adjacent acoustic events. The lack of such bilingual corpora further made model training difficult. Also, the English words were usually pronounced with Mandarin accent, so different from the standard English produced by native speakers. Here we present the ways to handle the code-switched nature of the course lectures considered.

A. Baseline

The simplest way to develop a bilingual recognition system is to use a phoneme set including all phonemes of the two languages for acoustic model construction, similarly a lexicon of all words needed for the two languages, and a language model based on the bilingual lexicon [18]. Such a system is certainly capable of recognizing bilingual speech, and is taken as the baseline here.

B. Bilingual Acoustic Modeling

The overall block diagram for improved bilingual acoustic modeling is in Fig. 3. We begin with a set of full state-tied triphone models based on the complete bilingual phoneme set, including all Mandarin phonemes plus English phonemes, trained with the complete bilingual training data. This is referred to as "Acoustic Models (Full)" in the block (A) at the upper left corner of Fig. 3, where blocks indicated by "HO" and "GE" represent triphone models with central phonemes in the host and guest languages respectively, although phonemes of different languages can appear in the context. To address the problem of lack of training data for the guest language, all acoustic units are classified into weak units (with insufficient training data, for example, guest language units) and strong units (with sufficient training data, for example, host language units) [19]. Here the acoustic unit refers to three possible levels: either a triphone model, an HMM state, or a Gaussian in an HMM state.

[2] The key terms never appearing in *Testing Set* were removed.

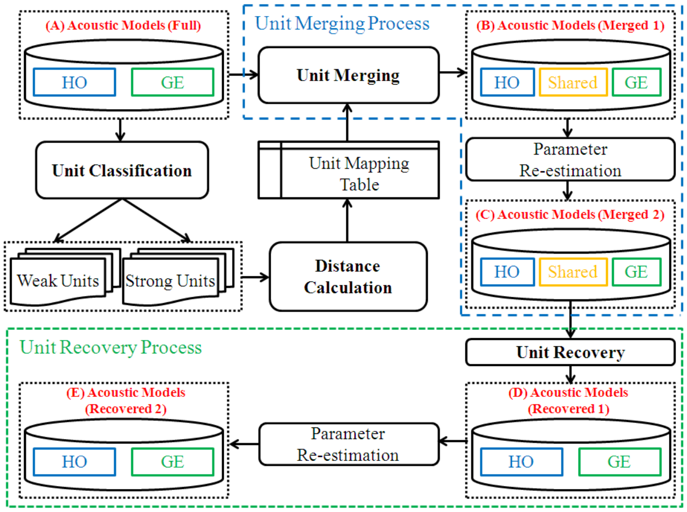


Fig. 3: *Proposed Approach for Bilingual Acoustic Modeling.*

With the lists of weak and strong acoustic units, distance calculation is performed between each weak unit (model, state, Gaussian) and all strong units within the same phonetic class uing symmetric KL divergence [18], [20]–[22]. This gives the mapping table at the upper middle of Fig. 3, based on which each weak unit (model, state, Gaussian) with too small training data is merged with a strong unit on the same level with minimum distance. In this way, the weak unit borrows the training data from the strong unit.

On the Gaussian level, merging is performed by combining the means and covariance matrices. For merging on the state level, every Gaussian in the state merges with another Gaussian in the corresponding state with minimum symmetric KL divergence. For merging on the model level, all states belonging to the model are respectively merged with its corresponding counterpart with state alignment estimated by state transition probabilities [20]. Such unit merging process produces a set of “shared units” as shown in the block (B) at the upper right corner of Fig. 3 as “Acoustic Models (Merged 1)”, in which the “shared units” are those produced when a weak unit is merged with the corresponding strong unit. The parameters for all “shared units” in “Acoustic Models (Merged 1)” are then re-estimated with maximum likelihood estimation in speaker dependent case or a cascade of Maximum Likelihood Linear Regression (MLLR) and Maximum a Posteriori (MAP) in speaker adaptation case. This gives the set of “Acoustic Models (Merged 2)” at the right middle of Fig. 3.

After the re-estimation, the merged units tend to be closer to the strong units than the weak units because the former dominate the data. Hence, we recover the merged units by copying all parameters from the merged units to be the respective units for both languages, and then applying an additional run of parameter re-estimation. This is illustrated in Fig. 3, where the recovery process gives the set of “Acoustic Models (Recovered 1)” in block (D) at the lower right corner which does not include the “shared units” any longer, and the parameter re-estimation gives the final set of “Acoustic Models (Recovered 2)” in block (E) at the lower left corner. In the last re-estimation process, parameters of all units for both languages can be estimated individually based on their own data, but with initial parameters estimated by the shared data when merged .

C. *Frame-level Guest Language Detection*

Here a frame-level guest language detector based on neural net and specially selected features is further integrated in the bilingual recognizer. For each speech frame o_t at time t, the guest language detector generates a posterior probability for the frame belonging to the guest language, $P(G|o_t)$. In the Viterbi search, if o_t is identified as in the guest language ($P(G|o_t) > 0.5$), its likelihood for each HMM state q_j for guest language models, $P(o_t|q_j)$, is boosted [23].

D. *Experiments*

For all the experiments in this subsection, the acoustic models used were all triphone models with state-clustering by decision trees. Two scenarios, speaker adaptation and speaker dependent acoustic modeling, were considered as mentioned in Section III. In both scenarios, the same lexicon and language model were used. The bilingual lexicon used for speech recognition included 2.1K English and 11.2K Chinese words. The 2.1K English words were selected from the slides and the reference transcriptions of *Training* and *Adaptation Sets* in Table I, which covered all of the English words in *Testing Set*. The 11.2K Chinese words included all commonly used Chinese characters taken as mono-character Chinese words and multi-character Chinese words discovered by PAT-Tree based approaches from a large corpus [24]. We used the Kneser-Ney trigram language model with a background model adapted with the transcriptions in *Training Set* in Table I. For the speaker adaptation scenario, the Mandarin speaker independent models were trained with 31.8 hours of the ASTMIC corpus of Mandarin read speech, and the English models with 29.7 hours of the EATMIC corpus of English read speech produced by Taiwanese speakers, and then adapted by *Adaptation Set* in Table I. The speaker dependent acoustic models were directly initialized and trained from *Training Set* in Table I. For evaluation, when aligning recognition results with the reference transcriptions, insertions, deletions and substitutions were evaluated respectively for each language and summed up for overall evaluation [23]. The basic unit for alignment and calculation was character for Mandarin [3] and word for English.

Overall accuracy and individual performance for both languages are summarized in Table II. The English accuracy is emphasized here because the English terms are usually the key terms, in addition to the lack of English training data. The results for both the standard speaker adaptation (SA, upper half) with cascaded MLLR and MAP, and speaker dependent (SD, lower half) models are reported in the table. In each

[3] Because in Mandarin Chinese different word sequences can correspond to the same character sequence, when evaluating the recognition performance, character accuracies are usually used instead of word accuracies.

Acoustic Models	Approach	Accuracy (%)		
		Mandarin	English	Overall
SA	Baseline	75.75	51.95	73.96
	BAM	76.72	58.06	75.32*
	BAM+GLD	76.77	58.51	75.40*†
SD	Baseline	83.62	61.87	81.99
	BAM	84.46	72.45	83.56*
	BAM+GLD	84.57	72.52	83.67*†

TABLE II: *Results for bilingual acoustic modeling (BAM) and the integration with guest language detection (BAM+GLD) for the scenario of speaker adaptation (SA) and speaker dependent (SD) modeling. The superscripts* * *and* † *on overall accuracies respectively indicate significantly better than the baseline system (Baseline) and the bilingual acoustic modeling (BAM).*

case, the results for the baseline system mentioned in Subsection IV-A without considering the bilingual characteristics (Baseline), for the bilingual acoustic modeling approaches described in Section IV-B with Gaussian unit merging and recovery (BAM), and for BAM plus the guest language detection in Subsection IV-C (BAM+GLD) are all listed in different rows. Pairwised t-test with significance level at 0.05 was also performed over the overall results (considering Mandarin and English jointly). The superscripts * and † on overall accuracies respectively indicate significantly better than the results in rows labeled "Baseline" and "BAM".

We can see from the table that the accuracies were improved by the bilingual acoustic modeling approach in all cases (BAM vs Baseline). With the unit merging and recovery, the English accuracy part was dramatically improved, while Mandarin accuracy was slightly improved as well. The integration with the guest language detection also offered further improvements (BAM+GLD vs BAM).

V. Semantic Analysis

We use a very popular approach for latent topic analysis: probabilistic latent semantic analysis (PLSA) [25]. Given the lecture corpus $\mathcal{L}$ here, PLSA obtained the probability of observing a word w given latent topic T_k, $P(w|T_k)$, and the mixture weight of topic T_k given document d, $P(T_k|d)$, where $\{T_k, k = 1, 2, ..., K\}$. For the multi-layer temporal structure mentioned in Section II-B, the documents d considered here can be either the utterance-level segments, the paragraphs, sections or chapters.

Several different measures can be obtained based on the PLSA model [26]. First of all, we can have a topic distribution given a word w,

$$P(T_k \mid w) = \frac{P(w \mid T_k)P(T_k)}{P(w)}, \tag{1}$$

where $P(w)$ can be obtained by estimating the probability of w in the corpus $\mathcal{L}$, and $P(T_k)$ can be estimated by averaging $P(T_k \mid d)$ over all the documents d in $\mathcal{L}$.

Given $P(T_k \mid w)$ in (1), latent topic entropy $E(w)$ for a word w [26] is defined:

$$E(w) = -\sum_{k=1}^{K} P(T_k \mid w) \log P(T_k \mid w). \tag{2}$$

Clearly, a lower $E(w)$ implies that the distribution of $P(T_k \mid w)$ is more focused on a smaller number of latent topics.

The latent topic significance of a word w with respect to a specific topic T_k [26] is defined as below:

$$S_w(T_k) = \frac{\sum_{d \in \mathcal{L}} f(w,d)P(T_k \mid d)}{\sum_{d \in \mathcal{L}} f(w,d)[1 - P(T_k \mid d)]}. \tag{3}$$

In the numerator of (3), the count of the given word w in each document d, $f(w, d)$, is weighted by the likelihood that the given topic T_k is addressed by the document d, $P(T_k \mid d)$, and then summed over all documents d in the corpus $\mathcal{L}$. The denominator is very similar except for latent topics other than T_k, or $P(T_k \mid d)$ in the numerator of (3) is replaced by $[1 - P(T_k \mid d)]$ in the denominator.

VI. Keyterm Extraction

We assume the key terms of a course are the fundamental elements of the semantics of the knowledge covered, so automatic extraction of key terms is very important. In key term extraction, supervised approaches can provide better performance than unsupervised approaches [27]. However, in real world application, a set of key terms as training examples for the supervised approaches is usually not available. Therefore, only unsupervised key term extraction are considered here, which is still a very challenging task today [28], [29].

Here the key terms considered include two types: key phrases (e.g. "hidden Markov model" and "information theory") and keywords (e.g. "perplexity"). TF-IDF is a good measure for identifying key phrases [30], [31], but it suffers from identifying some incomplete phrases (usually parts of key phrases) as key phrases. For example, TF-IDF may regard "hidden Markov" (an incomplete phrase) as a key phrase. To address this problem, in Section VI-A, right/left branching entropy is used to rule out the incomplete phrases. On the other hand, because a word may have different meanings in different context, but a phrase seldom has several meanings, identifying keywords are harder than key phrases, so more sophisticated approach is needed for identifying keywords. Therefore, in Section VI-B, we present an unsupervised two-stage approach for automatically selecting keywords, which realizes keyword extraction by considering various information from the Internet, the transcriptions and audio signals of the lectures including prosodic features [32].

A. Key Phrase Identification

The purpose here is to identify patterns of two or more words appearing together in the transcriptions of the lectures much more frequently than other sequences of words, so we can take them as candidates of key phrases such as "hidden Markov model" or "information theory". The approach proposed here is to use right/left branching entropy. The right branching entropy of a pattern u (two or more words), $H_r(u)$, is defined as

$$H_r(u) = -\sum_{a \in \mathcal{A}_u} p(a) \log p(a), \tag{4}$$

where u is the pattern of interest (e.g., "hidden Markov"), $\mathcal{A}_u$ is the set of all "child" patterns of u, found in the lecture transcriptions, or all patterns which are formed by appending a word after u (e.g., "hidden Markov model", "hidden Markov chain" for "hidden Markov"), a is an element of $\mathcal{A}_u$, and

$$p(a) = \frac{f(a)}{f(u)}, \tag{5}$$

where $f(u)$ and $f(a)$ are the frequency counts of u and a in the transcriptions respectively. Thus $p(a)$ is the probability of having a given u, and $H_r(u)$ is therefore the right branching entropy of u.

When a pattern "hidden Markov model" appears very frequently, most patterns of "hidden Markov" are all followed by the word "model" (so "hidden Markov" has a low $H_r(u)$), while the patterns of "hidden Markov model" are followed by many different words such as "is", "can", "to", "with"... (so "hidden Markov model" has a high $H_r(u)$). In this way we can use the right branching entropy $H_r(u)$ to identify the right boundary of a key phrase candidate (to the right of "model" rather than the right of "Markov" in the above example) by setting thresholds for $H_r(u)$.

Similarly we can define a left branching entropy $H_l(u)$ for each pattern u to be used similarly to identify the left boundary of a key phrase candidate (e.g. the left boundary of the phrase "hidden Markov model" is to the left of "hidden" rather than the left of "Markov", because "hidden" is preceded by many different words, while "Markov" is almost always preceded by "hidden").

In the test, we compute the average $H_r(u)$ and $H_l(u)$ for all possible patterns u, and then take those patterns u whose $H_r(u)$ and $H_l(u)$ are both higher than the average values to be the key phrase candidates. Then the key phrase candidates whose TF-IDF are higher than a threshold are selected as key phrases. The threshold can be determined by a development set.

B. Keyword Selection

All single words in the lecture transcriptions which is labeled as a "Noun" by a POS tagger and not in the stop word list are taken as candidates of keywords. With the framework shown in Fig. 4, in the first stage, all keyword candidates are ranked according to their topic coherence and term significance measures. In the second stage, based on the ranking in the first stage, pseudo-positive/-negative examples for keywords are selected to train an SVM classifier which decides the final keyword list.

1) First Stage – Candidate Ranking: The first reference for keyword ranking is the topic coherence. This is based on the observation that words having more coherent context are more likely to be keywords. For example, in the course related to speech processing, the keyword "perplexity" is usually surrounded by context regarding "language model", "entropy", etc; on the other hand, the word "equation" is not a keyword, it is usually surrounded by widely varying context. Hence, we evaluate the topic coherence of the context of each candidate keyword t.

The topic coherence of each candidate keyword t is evaluated as below. Given a database $\mathcal{D}$ (first consider the lecture transcriptions $\mathcal{L}$ as $\mathcal{D}$ here, although $\mathcal{D}$ will be generalized to other databases latter on), we train a PLSA model from $\mathcal{D}$ with a topic distribution $\{P(T_k|d), k = 1, 2, ..., K\}$. For each keyword candidate t, we then select the M documents out of $\mathcal{D}$ with the highest frequency counts of t as the contexts of t, $\mathcal{C}(t)$. The topic coherence for the keyword candidate t is then defined as the average pairwise cosine similarity for the documents d in $\mathcal{C}(t)$ as below:

$$h_{\mathcal{D}}(t) = \frac{\sum_{d,d' \in \mathcal{C}(t), d \neq d'} T(d,d')}{M(M-1)}, \tag{6}$$

where the subscript $\mathcal{D}$ in $h_{\mathcal{D}}(t)$ indicates that the topic coherence is based on the database $\mathcal{D}$, M is the size of the context $\mathcal{C}(t)$, and $T(d,d')$ is the cosine similarity between the PLSA topic distributions of d and d':

$$T(d,d') = \frac{\sum_{k=1}^{K} P(T_k \mid d) P(T_k \mid d')}{\sqrt{\sum_{k=1}^{K} P(T_k \mid d)^2} \sqrt{\sum_{k=1}^{K} P(T_k \mid d')^2}}. \tag{7}$$

We consider those keyword candidates t with higher $h_{\mathcal{D}}(t)$, are more likely to be keywords.

Latent topic entropy (LTE) described in (2) in Section V is certainly an important parameter for keyword ranking too. The latent topic entropy of a keyword candidate t, $E_{\mathcal{D}}(t)$, evaluated based on the PLSA model trained from the database $\mathcal{D}$ is therefore computed. The subscript $\mathcal{D}$ of $E_{\mathcal{D}}(t)$ indicates it is based on the database $\mathcal{D}$. A lower $E_{\mathcal{D}}(t)$ implies that t is more focused on less latent topics, or carries more topical information or salient semantics. With the term frequency jointly considered, the significance score of a keyword candidate t is defined as

$$s_{\mathcal{D}}(t) = \frac{\gamma f(t,\mathcal{D})}{E_{\mathcal{D}}(t)}, \tag{8}$$

where $f(t,\mathcal{D})$ is the frequency count of t in $\mathcal{D}$, and γ is a scaling factor.

The lecture transcriptions $\mathcal{L}$ can serve as the database $\mathcal{D}$ here, with each paragraph regarded as a document d. However, the information in the course lectures may be limited. This can be generalized using Google search engine and the Wikipedia. We use each keyword candidate t as the query to request the Google search engine, and the top M web pages returned by Google are regarded as $\mathcal{C}(t)$. In this way, the database $\mathcal{D}$ is approximately all the web pages on the Internet. Similarly, we also take all the Wikipedia pages as $\mathcal{D}$ by the search engine of Wikipedia.

Based on a database $\mathcal{D}$, each candidate keyword t is given a score $K_{\mathcal{D}}(t)$ by putting together (6) and (8),

$$K_{\mathcal{D}}(t) = h_{\mathcal{D}}(t) \cdot s_{\mathcal{D}}(t). \tag{9}$$

Finally, the candidate keywords are ranked according to the weighted sum of $K_{\mathcal{D}}(t)$ based on different databases $\mathcal{D}$.

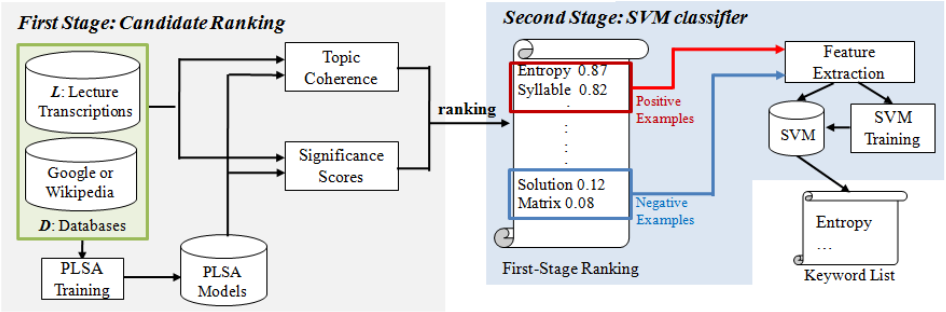


Fig. 4: *The framework of two-stage keyword extraction.*

2) Second Stage – SVM Classifier: From the candidate list ranked by the first stage, we simply assume the top M' candidates to be pseudo-positive examples and the bottom M' candidates to be pseudo-negative examples, and use these examples to train an SVM classifier. The features for SVM classifier training include prosodic features[4], lexical features (TF-IDF, POS tags, etc.), and semantic features (from PLSA, etc.) [32]. Finally, we use this SVM classifier to classify all the candidate keywords (including the selected examples) to decide whether they are keywords.

C. *Experiments*

Both the transcriptions generated by the baseline speaker adaptive models (row Baseline in the upper part of Table II) and manual transcriptions were used for key term extraction here. We used $1/10$ of the lecture transcriptions and the key terms included out of the 154 as the development set to tune the parameters including M (size of $\mathcal{C}(t)$ in (6)), γ in (8), and M' (number of SVM training examples), the weights for the sum of $K_{\mathcal{D}}(t)$ in (9) for different databases $\mathcal{D}$, and the parameter for SVM training. The number of PLSA topics was 25.

	Approach	Precision	Recall	F-measure
ASR	**English Phrases**	29.58	35.59	32.31
	Branching Entropy	58.54	81.36	68.09
Manual	**English Phrases**	41.27	44.07	42.62
	Branching Entropy	59.26	81.36	68.57

TABLE III: *Performance of key phrase extraction using ASR or manual transcriptions (%).*

The results (Precision, Recall and F1 measure) for key phrase extraction based on ASR or manual transcriptions are listed in Table III. In row labeled **English Phrases**, all the English noun phrases appearing in the transcriptions were taken as key phrase candidates, and the candidates whose TF-IDF higher than a threshold were selected as key phrase. The rows **Branching Entropy** are the results using the approach in Section VI-A. We found that the results in rows **Branching Entropy** was better than **English Phrases** in all cases. This shows that branching entropy could select better candidates than considering all the English noun phrases as candidates. We also find that results for ASR transcriptions were rather close to the manual case, probably because the phrase patterns had relatively high recognition accuracy. With ASR transcriptions, using branching entropies to find key phrase candidates yielded an F-measure of 68.09% in Table III, which implies the usefulness of this approach [5].

	Approach	Precision	Recall	F-measure
ASR	English Words	5.98	94.92	11.25
	TF-IDF	37.78	16.59	23.05
	K-means Exemplar	40.28	30.53	34.73
	Proposed	45.45	31.91	37.50
Manual	English Words	5.89	95.74	11.10
	TF-IDF	41.67	31.91	36.14
	K-means Exemplar	49.32	37.89	42.86
	Proposed	50.81	67.02	57.80

TABLE IV: *Performance of keyword extraction using ASR or manual transcriptions (%).*

The results (Precision, Recall and F1 measure) for keyword extraction using ASR or manual transcriptions are listed in Table IV compared to three baselines: English Words (all the noun in English appearing in the transcriptions were considered as keywords), TF-IDF (selecting $\bar{N}$ candidate keywords with the highest TF-IDF scores) and K-means exemplar (using K-means algorithm[6] to cluster candidate keywords based on latent topic distributions, and selecting exemplars of the clusters as keywords) [33], [34]. Considering all the English words as keywords obtain high recall rate but low precision rate. This is because most keywords were in English, but most of English words were not keywords. We find that the proposed approach outperformed both baselines in terms of all evaluation measures (Precision, Recall and F1 measure) for both ASR and manual transcriptions. F-measures of 37.50% and 57.80% respectively for ASR and manual transcriptions were obtained. Note that the results for key phrases in Table III are significantly better than keywords in Table IV. This implies

[4] pitch related, energy related, and duration related, since keywords are very often produced with wider pitch range, higher energy, and lower speed

[5] It is possible to apply the two-stage approach in Section VI-B for key phrase extraction, but this approach did not improve key phrase extraction in the experiments.

[6] setting $K = \bar{N}$

it is much more difficult to identify a simple word as a keyword than for a pattern as a key phrase.

VII. Keyterm Graph

Here we try to connect the related key terms into a key term graph on which each key term is a node, so the key term graph forms the backbone of the global semantic structure of the course lectures.

A. Approaches

We define a relationship function $R(t_i, t_j)$ for every two key terms t_i and t_j to describe the degree of relationship between them. The key term pairs t_i and t_j with $R(t_i, t_j)$ exceeding a threshold are considered as related, and linked on the key term graph. $R(t_i, t_j)$ can be one of the five functions $R_a(t_i, t_j)$ to $R_e(t_i, t_j)$ proposed below or their linear combinations.

(a) Co-occurrence Rate based on the lecture transcriptions $\mathcal{L}$:

$$R_a(t_i, t_j) = \frac{n(t_i, t_j)}{n(t_i) + n(t_j) - n(t_i, t_j)}, \quad (10)$$

where $n(t_i)$, $n(t_j)$ and $n(t_i, t_j)$ are the number of paragraphs containing t_i, t_j and both t_i and t_j.

(b) Word-level Context Coherence:

$$R_b(t_i, t_j) = \frac{2|C_w(t_i) \cap C_w(t_j)|}{|C_w(t_i)| + |C_w(t_j)|}, \quad (11)$$

where $C_w(t_i)$ is the word-level context of t_i, or the word set containing all words (excluding stop words) in the transcription paragraphs having the term t_i, and $|C_w(t_i)|$ is the number of distinct words in $C_w(t_i)$. Therefore, $R_b(t_i, t_j)$ is the dice co-efficient of the sets $C_w(t_i)$ and $C_w(t_j)$.

(c) Latent Topic Similarity based on the topic distribution given a term t_i, or $\{P(T_k|t_i), k = 1, ..., K\}$ as in (1):

$$R_c(t_i, t_j) = \frac{\sum_{k=1}^{K} P(T_k|t_i) P(T_k|t_j)}{\sqrt{\sum_{k=1}^{K} P(T_k|t_i)^2} \sqrt{\sum_{k=1}^{K} P(T_k|t_j)^2}}, \quad (12)$$

which is the cosine similarity between the topic distribution vectors very similar to (7).

(d) Similarity in Latent Topic Significance:

$$R_d(t_i, t_j) = \frac{\sum_{k=1}^{K} S_{t_i}(T_k) S_{t_j}(T_k)}{\sqrt{\sum_{k=1}^{K} S_{t_i}(T_k)^2} \sqrt{\sum_{k=1}^{K} S_{t_j}(T_k)^2}}, \quad (13)$$

which is parallel with (12), except that the topic distributions $P(T_k|t_i)$ in (1) used in (12) are replaced by the latent topic significances $S_{t_i}(T_k)$ in (3). In (13), those terms highly significant to the same topics are highly related.

(e) Inverse Normalized Google Distance (NGD):

$$R_e(t_i, t_j) = -NGD(t_i, t_j), \quad (14)$$

where the normalized Google Distance, $NGD(t_i, t_j)$, between two terms t_i and t_j is estimated inversely proportional to the possibility that t_i and t_j appear on the same web page obtained from Google search engine using t_i, t_j, as well as "t_i and t_j" as the queries [35]. The concepts of $R_a(t_i, t_j)$ and $R_e(t_i, t_j)$ are very similar, but $R_a(t_i, t_j)$ is based on the paragraphs of the course transcriptions, while $R_e(t_i, t_j)$ on the web pages.

B. Experiments

Here we constructed the key term graph based on the best results of automatically extracted key terms in Section VI with 57.80% of F1 measure in Table IV. We conduct 3-fold cross validation. The extracted key terms were first separated into 3 sets, roughly corresponding to the first, middle and last parts of the course. In each trial, 2 sets of key terms and their human-generated graphs were used as the development set to determine the respective thresholds for the relationship functions to be used for generating the key term graph for the remaining test set. This process was repeated 3 times. We used Graph Edit Distance (GED) [17] from the machine-generated graph to the human-generated graph as the evaluation measure. Smaller GED indicates better performance.

Table V shows the performance of the key term graphs evaluated by GED. The results based on the five relationship functions, $R_a(t_i, t_j)$ in (10) to $R_e(t_i, t_j)$ in (14), are respectively in rows (a) to (e). We find that the relationship functions based on semantic analysis, that is, latent topic similarity $R_c(t_i, t_j)$ in row (c) and similarity in latent topic significance $R_d(t_i, t_j)$ in row (d), yielded better results than other functions in rows (a), (b) and (e). This shows that the semantic analysis was really useful for keyterm graph construction. The relatively larger values of GED for rows (a) and (e) indicates that the related key terms did not necessarily co-occur in either the same course paragraphs or the same web pages, and the results in row (b) reveals that related key terms did not necessarily have coherent contexts. In addition, in row (f), we further weighted summed the scores in rows (c) and (d) with weights determined by the development set. We found that the integration of the two functions based on latent topics in rows (c) and (d) offered further improvements over the individuals (rows (f) vs (c), (d)). However, further integrating the results in (f) with (a), (b) and (e) did not provide any further improvements, probably because row (f) was much better than rows (a), (b) and (e).

In Fig 5, parts of the key term graphs generated by different relationship functions are displayed [7]. Fig 5 (a) is part of the reference key term graph. It is obvious that there are two groups of key terms. One group of key terms is related to speech recognition (at right hand side of Fig 5 (a)), and another group is more related to semantic analysis (at left hand side). Fig 5 (b) is part of the key term graph generated based on co-occurrence rate ($R_a(t_i, t_j)$ in (10)), or part of the result in row (a) in Table V. Compared with the reference key term graph, a large amount of related key terms could not be connected by co-occurrence rate. This is because some of the related key terms such as "MFCC" and "Hidden Markov Model" were introduced in different chapters, and thus had very low co-occurrence rate. The key term graphs generated by word-level context coherence ($R_b(t_i, t_j)$ in (11)) had the same issue as co-occurrence rate. Fig 5 (c) is part of the key term graph generated by integrating Latent Topic Similarity ($R_c(t_i, t_j)$ in (12)) and Similarity in Latent Topic Significance ($R_d(t_i, t_j)$ in (13)), or part of the result in row (f) in Table V. The graph

[7] We can not display the complete key term graph because there are too many key terms.

	Functions for Relation Evaluation	GED
(a)	$R_a(t_i, t_j)$: Co-occurrence Rate	0.182
(b)	$R_b(t_i, t_j)$: Word-level Context Coherence	0.212
(c)	$R_c(t_i, t_j)$: Latent Topic Similarity	0.136
(d)	$R_d(t_i, t_j)$: Similarity in Latent Topic Significance	0.152
(e)	$R_e(t_i, t_j)$: Inverse Normalized Google Distance	0.212
(f)	(c) + (d)	0.076

TABLE V: *Performance of keyterm graph construction based on different relationship functions in terms of graph edit distance (GED).*

in Fig 5 (c) is more similar to the reference graph in Fig 5 (a) than Fig 5 (b), which shows that semantic information can identify the related key terms with low co-occurrence rate.

VIII. Speech Summarization

While the key terms represent the fundamental semantic elements in the knowledge covered by the course and the key term graph represents the backbone of the global semantic structure, summaries can be obtained for the model in the multi-layer temporal structure (paragraphs, sections and chapters), which are significantly reduced temporal spoken knowledge based on the local semantic structure of the paragraphs, sections and chapters. They offer efficient ways in browsing the lectures. Therefore, here extractive summarization was performed on all paragraphs, sections and chapters. That is, given a set of utterance-level segments $\mathcal{X}$ (a paragraph, a section or a chapter), some segments $x \in \mathcal{X}$ are selected to form the summary $\mathcal{X}_{sum}$ of $\mathcal{X}$. Supervised summarization approaches [36], [37] have been successfully developed and used in lecture browsing systems based on sets of audio data and their reference summaries [7]. However, because the course content is usually on some specialized area, it is not easy to collect enough audio data in related domain, not to mention hiring experts understanding the content to produce reference summaries. Therefore, we assume unsupervised approaches are preferred for summarizing course lectures. Here a two-layer graph-based approach is used to summarize the segment sets by jointly considering the audio content and the corresponding slides. Similar two-layer graph-based approach has been proposed on multi-party meeting summarization [38], but it is used in a completely different way here. The original graph-based summarization approach [39] is described in Subsection VIII-A, and in Subsection VIII-B we introduce the two-layer approach.

A. Original Graph-based Approach

The basic idea of graph-based approach is that the segments similar to more segments in $\mathcal{X}$ is important, and the segments similar to the important segments are tend to be important as well. This can be formulated as a problem on a graph, in which all the segments x in $\mathcal{X}$ are nodes on the graph. The weights $T(x, x')$ of the directional edges from nodes x to x' ($x \rightarrow x'$) are the Okapi similarity between the transcriptions[8] of them [40]. Note that $T(x, x') = T(x', x)$, so there are two directional links with equal weights between each node pair $x \rightarrow x'$ and $x' \rightarrow x$, and only the node pairs with non-zero Okapi similarity are connected.

An importance score $M(x)$ is then assigned to each segment x based on the graph structure with the following equation:

$$M(x) = (1 - \lambda_S)M_0(x) + \lambda_S \sum_{x' \in I_{\mathcal{X}}(x)} M(x')T'(x', x), \quad (15)$$

where $I_{\mathcal{X}}(x)$ is the set of nodes in $\mathcal{X}$ connected to node x via incoming edges, and $T'(x', x)$ is the weight of the directional edge ($x' \rightarrow x$) normalized by the total weights over the outgoing edges of x':

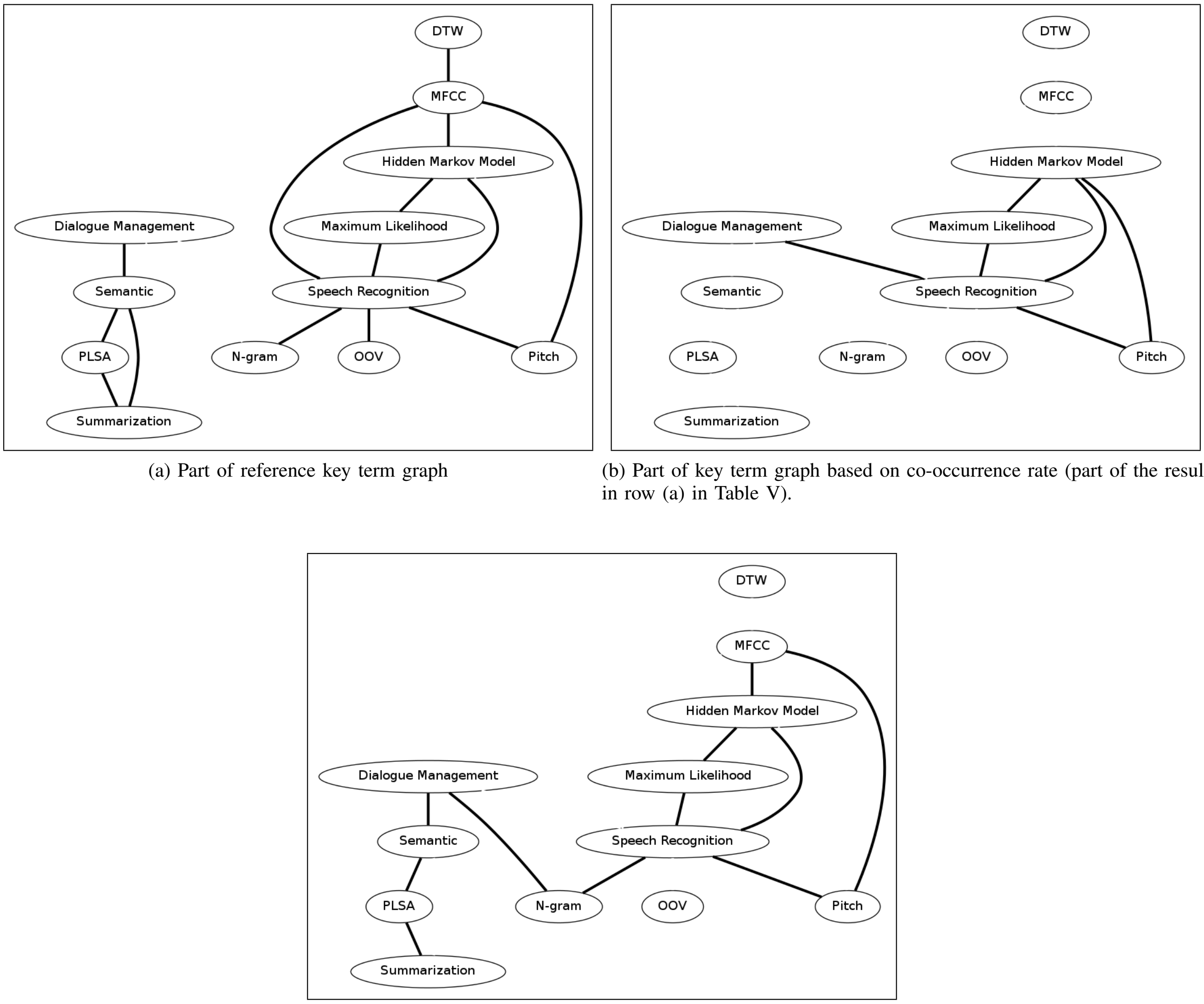
$$T'(x', x) = \frac{T(x', x)}{\sum_{x'' \in O_{\mathcal{X}}(x')} T(x', x'')}, \quad (16)$$

where $O_{\mathcal{X}}(x')$ are the set of nodes in $\mathcal{X}$ connected by outgoing edges of x'. Here (16) implies the score of node x' is distributed to all nodes x'' via the outgoing edges, so normalized by all outgoing edges; while the second term on the right of (15) implies the scores of all such nodes x' with an incoming edge linked to the node x flow to x. $M_0(x)$ in (15) are biases for segments x, which can be either uniform or obtained from some prior knowledge. λ_S is an interpolation weight between the two terms on the right of (15). Based on (15), the more x' is similar to x (or the higher the edge weight $T(x', x)$ is), the larger $M(x)$ is. We then used Maximum Marginal Relevance (MMR) to generate the summaries [41]. This approach selects in each iteration one utterance-level segment x from the segment set $\mathcal{X}$ to be added to the summary $\mathcal{X}_{sum}$ at the current iteration, which is the segment with the highest importance score $M(x)$, while adding minimum redundancy to the current summary $\mathcal{X}_{sum}$.

B. Two-layer Graph-based Approach

The slides of lecture courses are usually available. Compared with the content of the audio, the slides of lecture courses are succinct and well-structured, so they are very good resources to enhance the summarization of the lecture courses. Here the two-layer graph-based approach is used to jointly consider the utterance-level segments with their corresponding slides. As shown in Fig. 6, the graph has two layers of nodes. One layer of nodes is segments x in the segment set $\mathcal{X}$, and the other one is sentences s in the corresponding slides $\mathcal{S}$. The segments or sentences with Okapi similarity larger

[8] The function words are removed.



(a) Part of reference key term graph

(b) Part of key term graph based on co-occurrence rate (part of the result in row (a) in Table V).

(c) Part of key term graph based on the integration of latent topic similarity and similarity in latent topic significance (part of the result in row (f) in Table V)

Fig. 5: Part of the reference key term graph and the key term graphs generated by different relationship functions.

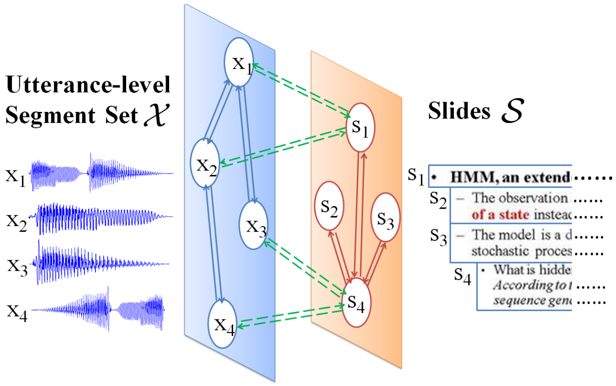


Fig. 6: *Two-layer Graph-based Approach.*

than zero are connected to each other. The basic idea of the two-layer approach is that the segments connect to the important sentences on the graph are important, and on the other hand, the sentences connecting to important segments are important as well. With the two-layer graph-based approach, the importance estimations of segments and sentences in slides are jointly considered and can reinforce each other.

Based on the two-layer graph, segments x and sentences s are assigned a set of new importance scores $M'(x)$ and $M'(s)$. The importance scores $M'(x)$ for segments x satisfy the following equation:

$$M'(x) = (1 - \lambda_S)M_0(x) + \lambda_S \sum_{x' \in I_{\mathcal{X}}(x)} M''(x')T'(x', x), \tag{17}$$

which is parallel to (15), except that in the second term at the right hand side, another scores $M''(x')$ are used rather than $M'(x')$. The score $M''(x')$ of segment x' depends on the importance scores $M'(s)$ of the sentences s in the slides:

$$M''(x') = \sum_{s \in I_{\mathcal{S}}(x')} M'(s)T'(s,x'), \tag{18}$$

where $I_{\mathcal{S}}(x')$ is the set of sentences in slides $\mathcal{S}$ connected to segment x' via incoming edges. $T'(s,x')$ is the weight of the directional edge $(s \to x')$ normalized by the total weights over the outgoing edges of s connecting to nodes belonging to $\mathcal{X}$:

$$T'(s,x') = \frac{T(s,x')}{\sum_{x'' \in O_{\mathcal{X}}(s)} T(s,x'')}, \tag{19}$$

where $T(s,x')$ is the Okapi similarity between the sentence s and the transcription of x', and $O_{\mathcal{X}}(s)$ are the set of segments in $\mathcal{X}$ connected by outgoing edges of sentence s. Based on (17) and (18), a segment can have large $M'(x)$ if connected by other segments with large $M''(x')$, and a segment x' have large $M''(x')$ when connected by important sentences s with large $M'(s)$. The importance of sentences $M'(s)$ is defined in a similar way as $M'(x)$ in (17).

$$M'(s) = (1-\lambda_S)M_0(s) + \lambda_S \sum_{s' \in I_{\mathcal{S}}(s)} M''(s')T'(s',s), \tag{20}$$

$$M''(s') = \sum_{x \in I_{\mathcal{X}}(s')} M'(x)T'(x,s') \tag{21}$$

Equations (20) and (21) are parallel with (17) and (18), except that the roles of segments and sentences are reversed. By searching for a set of $M'(x)$ and $M'(s)$ satisfying (17), (18), (20) and (21), the importance of the utterance-level segments and sentences in the slides are jointly estimated. MMR is finally used to select the segments to form the summaries based on the importance scores $M'(x)$ in (17).

C. Experiments

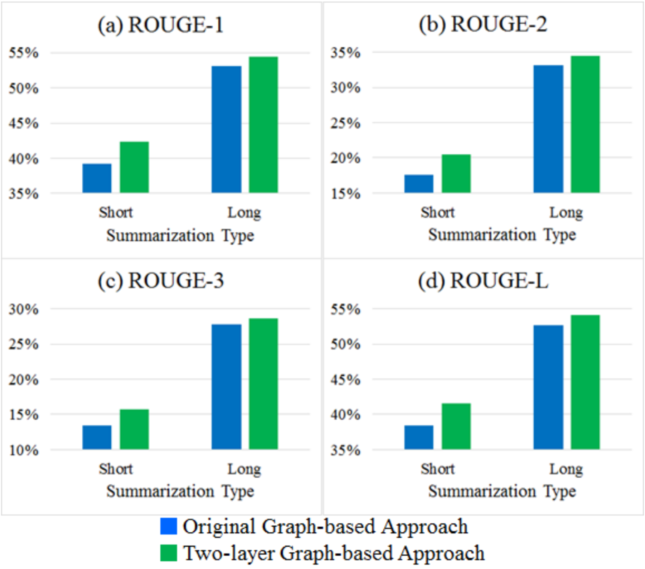


Fig. 7: *ROUGE-1, 2, 3 and ROUGE-L F-measures for original and two-layer graph-based approach with different types of summarization (Short and Long).*

To evaluate the performance of the automatically generated summaries, the ROUGE-N ($N = 1, 2, 3$) and ROUGE-L F-measures from the package ROUGE [42] were used. The results for the 40 sections with references in Section III were reported here. λ_S in (15), (17) and (20) was set to be 0.85. In all the experiments, we simply set the prior $M_0(x)$ in (15) and (17) to be $1/|\mathcal{X}|$, where $|\mathcal{X}|$ is the number of segments in the segment set $\mathcal{X}$ to be summarized, while $M_0(s)$ in (20) was set to be $1/|\mathcal{S}|$, where $1/|\mathcal{S}|$ is the number of sentences in the corresponding slides. We used the baseline speaker adaptive models (row Baseline in the upper part of Table II) to transcribe the audio of the 40 sections. Fig. 7 (a) to (d) respectively shows the results of ROUGE-1, 2, 3 and ROUGE-L F-measures for the 40 sections with references. In each case the two groups of bars are for the results of short (10% summarization ratio) and long summaries (30% summarization ratio), and in each group the blue bar is for original graph-based approach in Subsection VIII-A, while the green bar for two-layer graph-based approach in Subsection VIII-B. In all cases in Fig. 7, we see the two-layer graph-based approach improved the performance over the original approach regardless of the evaluation measures and summarization types.

IX. Spoken Content Retrieval

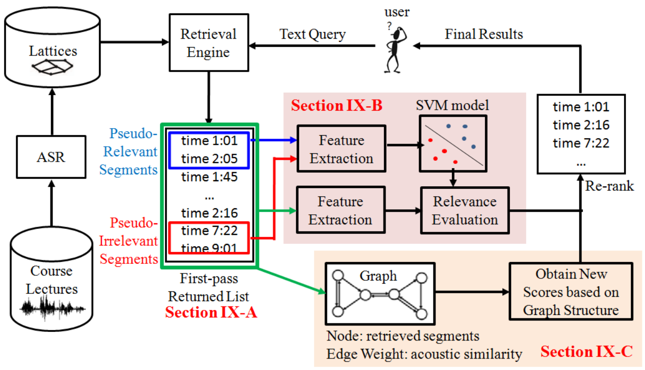


Fig. 8: *Framework for Pseudo-relevance Feedback (PRF).*

Given the semantic structure, key term graph and summaries, efficient and accurate retrieval of the spoken content in the lectures is finally the key element enabling the users to navigate across the course content for personalized learning. Here we focus on a specific task, in which the query is a term in text, and the system in to return utterance-level segments including the query term. The approaches presented below can be equally applied on retrieving paragraphs, sections and chapters, but here we only mention those for the utterance-level segments for simplicity. The discussions below may also be generalized to other spoken content retrieval tasks such as spoken queries or semantic retrieval, but they are out of the scope here.

Most conventional spoken content retrieval techniques were applied on top of ASR output such as lattices with performance inevitably depending on ASR accuracy. Because it is difficult to obtain acoustic and language models robust enough for recognizing spontaneous course lectures on specific subject domains, here we present different techniques for retrieving course lectures which are less constrained by recognition accuracy with the framework shown in Fig. 8 [43], [44]. When a query term is entered, the system generates first-pass retrieved results from the lattices. Two pseudo-relevance feedback (PRF) approaches are then applied, the one based on SVM in the middle of the figure, and the one based on graphs in the lower part of the figure. The results of the two approaches are then integrated.

A. First-pass Retrieval

Each utterance-level segment x in the course lecture archive is transcribed into a lattice off-line. When the query Q is entered, all segments x in the archive are ranked based on the widely used relevance score $S(Q,x)$, or the expected occurrence count of query Q obtained based on the acoustic and language model scores in the lattice of x [45]–[49]. This generates the first-pass retrieval results as in the middle left of Fig. 8. This list is not shown to the user. For simplicity, we assume the query Q is a single word, and the arcs in the lattices are word hypotheses. Extension to longer queries and subword lattices is trivial [44].

B. Pseudo-relevance Feedback based on SVM

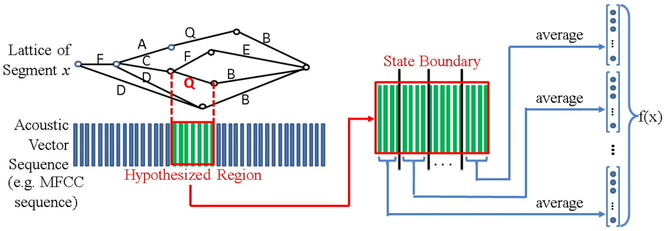


Fig. 9: *Feature vector representations. Left half: the definition of a "hypothesized region" in the lattice of segment x for the query term Q. Right half: the feature vector $f(x)$.*

As shown in the middle of Fig. 8, we select some segments in the first-pass retrieval results, assume they are pseudo-relevant and -irrelevant, and take them as positive and negative examples in training a support vector machine (SVM), with which the segments in the first-pass results are re-ranked [43].

To train an SVM model, each segment x should be represented by a feature vector $f(x)$. We first define the "hypothesized region" for a spoken segment x and a query Q to be the part of the acoustic vector (e.g., MFCC) sequence corresponding to a word arc in the lattice whose hypothesis is exactly Q with the highest posterior probability, as shown in the left half of Fig. 9. In the right half of Fig. 9, this hypothesized region is divided into a sequence of divisions based on the HMM state boundaries obtained during the lattice construction. Each division is then represented by the average of the acoustic vectors in it. All these averaged vectors in a hypothesized region are then concatenated to form the feature $f(x)$. For l-state phoneme HMMs and a query term Q including m phonemes, the dimensionality of such a feature vector $f(x)$ is $m \times l$ times the dimensionality of the acoustic vectors. The feature vector $f(x)$ thus capsules the acoustic characteristics of the hypothesized region of x.

Then as shown in the middle of Fig. 8, some segments in the first-pass retrieved list are respectively taken as positive and negative examples to train an SVM model. Each segment x in the list is first compared with the groups of top- and bottom-ranked segments and obtains a confidence measure for containing the query Q (those similar to many top segments and dissimilar to more bottom segments have higher confidence measures). In this way, we select positive and negative examples based on these confidence measures with such selected example having an estimated confidence [43].

The SVM training is slightly modified to consider the above confidence measure, so examples with higher confidence are weighted higher [43]. This SVM then classifies all segments in the first-pass results by giving each segment x a value which tends to be larger when x is relevant and vice versa. This value is then linearly normalized into a real number $R(x)$ between 0 and 1. The new relevance score $S_1(Q,x)$ for re-ranking the segment x is then obtained by integrating the original relevance score $S(Q,x)$ in Subsection IX-A with $R(x)$,

$$S_1(Q,x) = S(Q,x)^{1-\delta_R} R(x)^{\delta_R}, \tag{22}$$

where δ_R is a weight parameter.

C. Pseudo-relevance Feedback based on Graphs

The basic idea here is very similar to segment scoring using graphs for summairzation in Section VIII-A. If the hypothesized region of a segment is very similar to many other segments judged to include the query Q in the first-pass retrieval, it may have a higher probability to include the query Q. Therefore, we construct a graph with each segment x in the first pass results being a node. The similarity $T_R(x,x')$ between two segments x and x' (the weight for edge $x \rightarrow x'$) can be estimated based on the DTW distance between the hypothesized regions (defined on the left of Fig. 9) of them plus a transformation (larger distance implying smaller similarity). Then the graph is pruned such that each node (segment) x is connected to only K_R segments x' with the highest incoming edge weight $T_R(x,x')$.

The rest is similar to Section VIII-A. Each segment x is assigned a new score $S^G(x)$,

$$S^G(x) = (1-\lambda_R)S(Q,x) + \lambda_R \sum_{x' \in IN(x)} S^G(x')T'_R(x',x), \tag{23}$$

where $T'_R(x',x)$ is the edge weight $T_R(x',x)$ normalized over the outgoing edges of segment x':

$$T'_R(x',x) = \frac{T_R(x',x)}{\sum_{x'' \in OUT(x')} T_R(x',x'')}. \tag{24}$$

Equations (23) and (24) are exactly in parallel with (15) and (16). λ_R in (23) is an interpolation weight. Here (23) implies $S^G(x)$ depends on two factors, the original scores $S(Q,x)$ in the first term and the scores propagated from similar segments in the second term. $S^G(x)$ is then integrated with the original relevance score $S(Q,x)$ for re-ranking as

$$S_2(Q,x) = S(Q,x)^{1-\delta'_R} S^G(x)^{\delta'_R}, \tag{25}$$

where δ'_R is a weight parameter.

D. Integration

The approaches in the Subsection IX-B and IX-C can be integrated as

$$S_3(Q,x) = S_1(Q,x)^{1-\delta''_R} S_2(Q,x)^{\delta''_R}, \tag{26}$$

where δ''_R is another weight parameter. The final results shown to the users are ranked according to $S_3(Q,x)$.

E. Experiments

Mean average precision (MAP) was used as the performance measure. Pair-wise t-test with significance level at 0.05 was used for significance test for improvements. δ_R, δ'_R and δ''_R in (22), (25) and (26) were respectively set to be 0.9, 0.9 and 0.5, and all the remaining parameters were determined by 4-fold cross validation. The testing queries were separated into 4 parts. In each trial, one part was used as the development query set for parameter tuning while the other three parts tested.

	SA	**SD**
(1)first pass	0.7962	0.8520
(2)SVM	0.8153*	0.8648
(3)graph	0.8357*†	0.8719*
(4)SVM+graph	0.8439*†‡	0.8783*†‡

TABLE VI: *Experimental Results yielded by pseudo-relevance feedback (PRF) in terms of Mean Average Precision (MAP). The superscripts *, † and ‡ respectively indicate significantly better than the first-pass results, SVM and the graph-based approach.*

The experimental results are shown in Table VI. We used the baseline speaker adaptive models (row Baseline in the upper part of Table II) and baseline speaker dependent models (row Baseline in the lower part of Table II) to transcribe all utterance-level segments in *Testing set* in Table I into two sets of lattices with beam width 50. The results based on the two sets of lattices are respectively in columns **SA** (speaker adaptive) and **SD** (speaker dependent) in Table VI. In Table VI, row (1) is the MAP scores of the first-pass retrieval results (Subsection IX-A), while those using SVM (Subsection IX-B), graphs (Subsection IX-C) and both (Subsection IX-D) are respectively in rows (2), (3) and (4). The superscripts *, † and ‡ indicate respectively significantly better than the first-pass results, SVM and the graphs. We find that both SVM and graphs improved the performance (rows (2),(3) vs (1)). Each of them showed strengths over the other. SVM took advantages of the discriminative capabilities of SVM, whereas the graphs considered the global acoustic similarities among all segments rather than the individuals. Hence, the integration in row (4) yielded further improvement over the individuals.

X. Prototype System

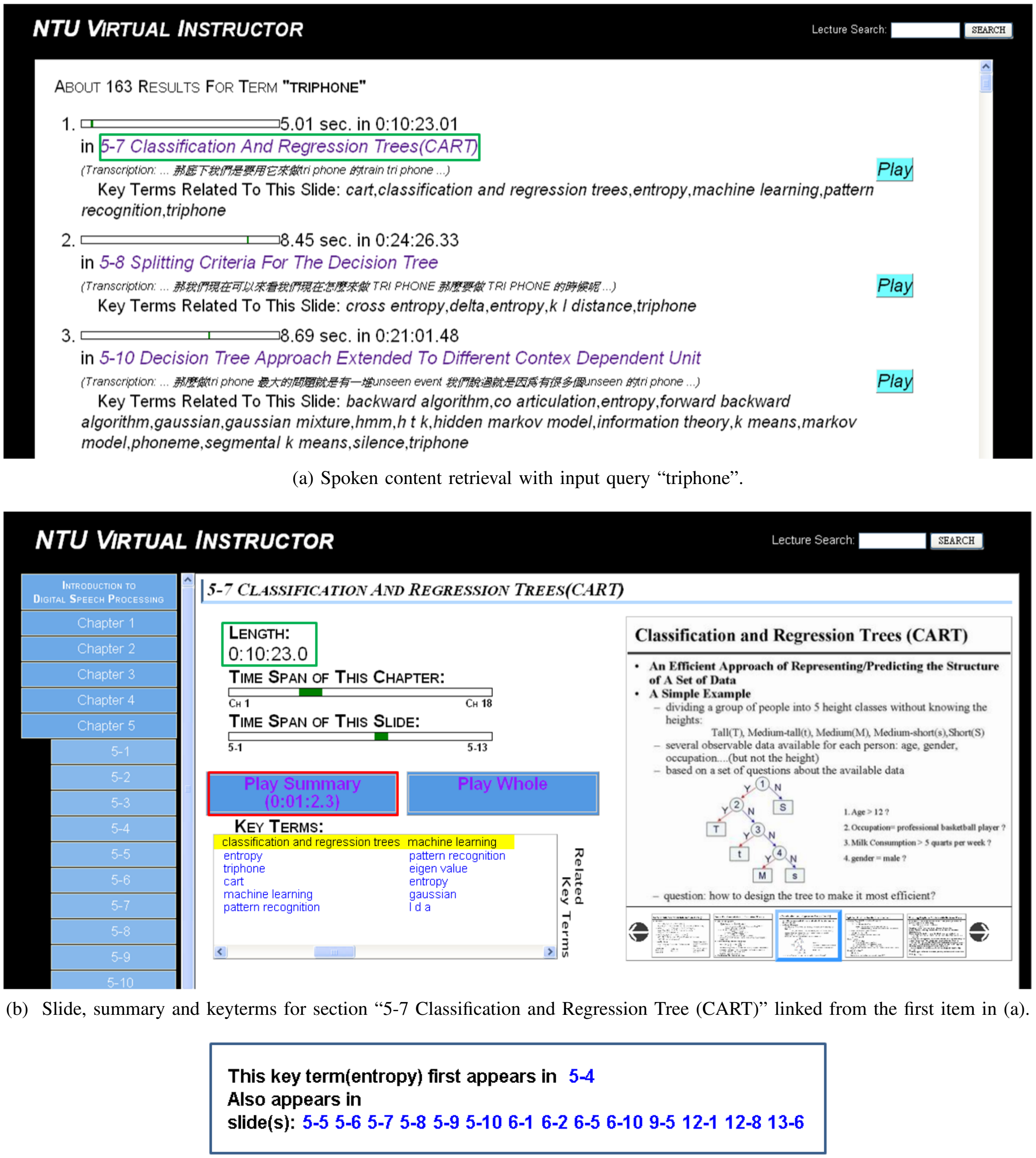
A preliminary prototype system has been successfully developed at National Taiwan University (NTU), referred to as NTU Virtual Instructor [10]. The first version of the system was completed in 2009 [9], while this paper presents the technologies used in its latest version. It is based on a course of “Digital Speech Processing” offered in NTU in 2006. Fig. 10 are example screenshots for learner/system interactions.

In Fig. 10 (a), the learner typed the query “triphone” in the blank at the upper right corner. As shown at the upper left corner, the spoken content retrieval system found 163 utterance-level segments containing the query term “triphone”, ranked by confidences (only top three shown here). The first segment was shown to belong to the section with slide title “5-7 Classification and Regression Tree (CART)” (“5-7” means the 7-th section in chapter 5), and the key terms in this section (such as CART, entropy, etc.) were also listed to help the learner judge if he was interested and able to understand this section. The learner could click the button “Play” and listen to the course starting from the returned segment, or click the link “5-7 Classification and ......” (in the green frame) to jump to the section 5-7 as the screenshot in Fig. 10 (b).

In Fig. 10 (b), the learner found that this section was 10 minutes and 23 seconds long (in the green frame), but he could click the bottom “Play Summary” (with the red edges) to listen to a summary of only 1 minute and 2 seconds long. In addition to the slide, the learner also saw a list of key terms extracted in this section in a yellow bar including “Classification and Regression Tree” and “Machine learning”. Other key terms below each key term in the yellow bar were those connected to the key term in the yellow bar on the key term graph (e.g. “entropy”, “triphone”, etc. below “Classification and Regression Tree” were those connected to “Classification and Regression Tree”[9]). If the learner clicked the key term “entropy”, the system then showed all sections in the temporal structure containing this key term including where the key term appeared the first time as an example learning path recommended by the system as shown in Fig. 10 (c). Therefore, the learner can choose to learn more about “entropy” sequentially from the beginning or towards more advanced topics if needed.

The subjective user tests were conducted to gauge the efficiency of the proposed spoken knowledge organization approaches for course lectures. Each subject was asked to answer ten questions specifically designed based on the content of the target course, which are listed in Table VII. The subjects used either the prototype system in Fig. 10 or a baseline system to help them find the answers (but they could not consult any other materials). The baseline system simply put the course on the internet without spoken knowledge organization as normal MOOC platforms or course warehouses. The interface of the baseline system was exactly the same as Fig. 10 (b), except that there were no key terms, summaries and search blank. Ten graduate students of National Taiwan University who had taken the target course participated in the test. To remove the

[9] Classification and regression tree is used to tie triphones here, and entropy was used as the criterion for node splitting.



ABOUT 163 RESULTS FOR TERM "TRIPHONE"

1. 5.01 sec. in 0:10:23.01
in *5-7 Classification And Regression Trees(CART)*
(Transcription: ... 那底下我們是要用它來做tri phone 的train tri phone ...)
Key Terms Related To This Slide: *cart,classification and regression trees,entropy,machine learning,pattern recognition,triphone*

2. 8.45 sec. in 0:24:26.33
in *5-8 Splitting Criteria For The Decision Tree*
(Transcription: ... 那我們現在可以來看我們現在怎麼來做 TRI PHONE 那麼要做 TRI PHONE 的時候呢 ...)
Key Terms Related To This Slide: *cross entropy,delta,entropy,k l distance,triphone*

3. 8.69 sec. in 0:21:01.48
in *5-10 Decision Tree Approach Extended To Different Contex Dependent Unit*
(Transcription: ... 那麼做tri phone 最大的問題就是有一堆unseen event 我們說過就是因為有很多個unseen 的tri phone ...)
Key Terms Related To This Slide: *backward algorithm,co articulation,entropy,forward backward algorithm,gaussian,gaussian mixture,hmm,h t k,hidden markov model,information theory,k means,markov model,phoneme,segmental k means,silence,triphone*

(a) Spoken content retrieval with input query "triphone".

5-7 CLASSIFICATION AND REGRESSION TREES(CART)

LENGTH:
0:10:23.0
TIME SPAN OF THIS CHAPTER:
CH 1 CH 18
TIME SPAN OF THIS SLIDE:
5-1 5-13

KEY TERMS:

classification and regression trees	machine learning
entropy	pattern recognition
triphone	eigen value
cart	entropy
machine learning	gaussian
pattern recognition	l d a

Classification and Regression Trees (CART)

- **An Efficient Approach of Representing/Predicting the Structure of A Set of Data**
- **A Simple Example**
  - dividing a group of people into 5 height classes without knowing the heights:
    Tall(T), Medium-tall(t), Medium(M), Medium-short(s),Short(S)
  - several observable data available for each person: age, gender, occupation....(but not the height)
  - based on a set of questions about the available data

1. Age > 12 ?
2. Occupation= professional basketball player ?
3. Milk Consumption > 5 quarts per week ?
4. gender = male ?

- question: how to design the tree to make it most efficient?

(b) Slide, summary and keyterms for section "5-7 Classification and Regression Tree (CART)" linked from the first item in (a).

This key term(entropy) first appears in 5-4
Also appears in
slide(s): 5-5 5-6 5-7 5-8 5-9 5-10 6-1 6-2 6-5 6-10 9-5 12-1 12-8 13-6

(c) Example learning path for the key term "Entropy" recommended by the system.

Fig. 10: Example screenshots of the prototype system.

bias from the users, the users were arranged into two groups. The users in group one answered questions number one to five by interacting with the baseline system and answered questions number six to ten by the prototype system, while the users in group two did the opposite. In this way, each user used both the baseline and prototype systems when answering the ten questions. The amount of time (seconds) required to answer each question is in Table VII. We found that the users can answer all of the questions faster when using the prototype system, except question number one ("List three HMM basic problems"). Because question number one is a very basic question for the target course, some of the users can answer this question directly without consulting the systems. This is why the prototype system was not very helpful for this question. The last row in Table VII shows the average amount of time required to answer the questions. The users took 134 seconds in average to answer a question with the baseline system, but only 49 seconds with the prototype system. The prototype system helped the users to answer the questions more than two times faster.

XI. CONCLUSION

This paper presents a new approach of organizing the spoken knowledge covered by course lectures for efficient

Questions	Baseline system	Prototype system
1. List three HMM basic problems.	47	53
2. List three speaker adaptation approaches.	114	68
3. List five language model smoothing approaches.	41	29
4. What does "CART" stand for?	217	8
5. When and which conference was the first paper of PLSI published?	310	143
6. List the numbers of Mandarin syllables, initials and finals.	83	67
7. List three DSR models.	93	12
8. List two evaluation measures for information retrieval.	99	35
9. What does "MFCC" stand for?	35	33
10. When was the first paper of Spectrum Subtraction published?	299	46
Average	134	49

TABLE VII: The amount of time (seconds) required to answer the questions using baseline and prototype systems.

personalized learning with multiple ways of learner/content interactions. Key terms for the course content are automatically extracted and connected into a key term graph to form the backbone of the global semantic structure of the course. Summaries are generated for each paragraph, section and chapter forming the multi-layer temporal structure interconnected with the above structure. Spoken content retrieval together with the above structured knowledge jointly offer an efficient way for the learner to navigate across the course content and develop personalized learning paths. A preliminary prototype system is successfully developed based on a course offered in Mandarin-English code-mixed speech.

References

[1] G. Tur and R. DeMori, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley & Sons Inc, 2011, ch. 15, pp. 417–446.

[2] M. Larson and G. J. F. Jones, "Spoken content retrieval: A survey of techniques and technologies," *Found. Trends Inf. Retr.*, vol. 5, pp. 235–422, 2012.

[3] J. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Interspeech*, 2007.

[4] I. Szoke, J. Cernocky, M. Fapso, and J. Zizka, "Speech@FIT lecture browser," in *SLT*, 2010.

[5] K. Riedhammer, M. Gropp, and E. Noth, "The FAU video lecture browser system," in *SLT*, 2012.

[6] R. Rose, A. Norouzian, A. Reddy, A. Coy, V. Gupta, and M. Karafiat, "Subword-based spoken term detection in audio course lectures," in *ICASSP*, 2010.

[7] Y. Fujii, K. Yamamoto, N. Kitaoka, and S. Nakagawa, "Class lecture summarization taking into account consecutiveness of important sentences," in *Interspeech*, 2008.

[8] S. Togashi and S. Nakagawa, "A browsing system for classroom lecture speech," in *Interspeech*, 2008.

[9] S.-Y. Kong, M.-R. Wu, C.-K. Lin, Y.-S. Fu, and L.-S. Lee, "Learning on demand – course lecture distillation by information extraction and semantic structuring for spoken documents," in *ICASSP*, 2009.

[10] *http://speech.ee.ntu.edu.tw/~RA/lecture/*.

[11] A. Park, T. J. Hazen, and J. R. Glass, "Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling," in *ICASSP*, 2005.

[12] T. Kawahara, N. Katsumaru, Y. Akita, and S. Mori, "Classroom note-taking system for hearing impaired students using automatic speech recognition adapted to lectures," in *Interspeech*, 2010.

[13] Y. Fujii, K. Yamamoto, and S. Nakagawa, "Improving the readability of class lecture ASR results using a confusion network," in *Interspeech*, 2010.

[14] H. Yamazaki, K. Iwano, K. Shinoda, S. Furui, and H. Yokota, "Dynamic language model adaptation using presentation slides for lecture speech recognition," in *Interspeech*, 2007.

[15] J. Miranda, J. P. Neto, and A. W. Black, "Improving ASR by integrating lecture audio and slides," in *ICASSP*, 2013.

[16] S.-C. Hsu, "Topic segmentation on lecture corpus and its application," Master's thesis, National Taiwan University, 2008.

[17] X. Gao, B. Xiao, D. Tao, and X. Li, "A survey of graph edit distance," *Pattern Anal. Appl.*, vol. 13, pp. 113–129, 2010.

[18] N. T. Vu, D.-C. Lyu, J. Weiner, D. nic Telaar, T. Schlippe, F. Blaicher, E.-S. Chng, T. Schultz, and H. Li, "A first speech recognition system for Mandarin-English code-switch conversational speech," in *ICASSP*, 2012.

[19] T. Niesler, "Language-dependent state clustering for multilingual acoustic modeling," in *Speech Communication*, 2007.

[20] C.-F. Yeh, L.-C. Sun, C.-Y. Huang, and L.-S. Lee, "Bilingual acoustic modeling with state mapping and three-stage adaptation for transcribing unbalanced code-mixed lectures," in *ICASSP*, 2011.

[21] Y. Qian and J. Liu, "Phone modeling and combining discriminative training for Mandarin-English bilingual speech recognition," in *ICASSP*, 2010.

[22] H. Cao, T. Lee, and P. Ching, "Cross-lingual speaker adaptation via gaussian component mapping," in *Interspeech*, 2010.

[23] C.-F. Yeh, A. Heidel, H.-Y. Lee, and L.-S. Lee, "Recognition of highly imbalanced code-mixed bilingual speech with frame-level language detection based on blurred posteriorgram," in *ICASSP*, 2012.

[24] C.-F. Yeh, C.-Y. Huang, L.-C. Sun, and L. Lee, "An integrated framework for transcribing Mandarin-English code-mixed lectures with improved acoustic and language modeling," in *ISCSLP*, 2010.

[25] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, 1999.

[26] S.-Y. Kong and L.-S. Lee, "Semantic analysis and organization of spoken documents based on parameters derived from latent topics," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 1875 –1889, 2011.

[27] Y.-N. Chen, Y. Huang, S.-Y. Kong, and L.-S. Lee, "Automatic key term extraction from spoken course lectures using branching entropy and prosodic/semantic features," in *SLT*, 2010.

[28] F. Liu, D. Pennell, F. Liu, and Y. Liu, "Unsupervised approaches for automatic keyword extraction using meeting transcripts," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

[29] F. Liu, F. Liu, and Y. Liu, "Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion," in *SLT*, 2008.

[30] E. D'Avanzo, B. Magnini, and A. Vallin, "Keyphrase extraction for summarization purposes: The LAKE system at DUC-2004," in *Proceedings of the 2004 document understanding conference*, 2004.

[31] X. Jiang, Y. Hu, and H. Li, "A ranking approach to keyphrase extraction," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009.

[32] Y.-N. Chen, Y. Huang, H.-Y. Lee, and L.-S. Lee, "Unsupervised two-stage keyword extraction from spoken documents by topic coherence and support vector machine," in *ICASSP*, 2012.

[33] Y.-N. Chen, Y. Huang, S.-Y. Kong, and L.-S. Lee, "Automatic key term extraction from spoken course lectures using branching entropy and prosodic/semantic features," in *SLT*, 2010.

[34] Z. Liu, P. Li, Y. Zheng, and M. Sun, "Clustering to find exemplar terms for keyphrase extraction," in *EMNLP*, 2009, pp. 257–266.

[35] R. Cilibrasi and P. Vitanyi, "The Google similarity distance," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, pp. 370–383, 2007.

[36] J. Zhang, H. Y. Chan, P. Fung, and L. Cao, "A comparative study on speech summarization of broadcast news and lecture speech," in *Interspeech*, 2007.

[37] S. Xie, D. Hakkani-Tur, B. Favre, and Y. Liu, "Integrating prosodic features in extractive meeting summarization," in *ASRU*, 2009.

[38] Y.-N. Chen and F. Metze, "Two-layer mutually reinforced random walk for improved multi-party meeting summarization," in *SLT*, 2012, pp. 461–466.

[39] Y.-N. Chen, Y. Huang, C.-F. Yeh, and L.-S. Lee, "Spoken lecture summarization by random walk over a graph constructed with automatically extracted key terms," in *Interspeech*, 2011.

[40] S. Robertson, S. Walker, M. Beaulieu, and P. Willett, "Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive track," *In*, vol. 21, pp. 253–264, 1999.

[41] S. Xie and Y. Liu, "Using corpus and knowledge-based similarity measure in maximum marginal relevance for meeting summarization," in *ICASSP*, 2008.

[42] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Workshop on Text Summarization Branches Out*, 2004.

[43] H.-Y. Lee and L.-S. Lee, "Enhanced spoken term detection using support vector machines and weighted pseudo examples," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 6, pp. 1272–1284, 2013.

[44] H.-Y. Lee, P.-W. Chou, and L.-S. Lee, "Open-vocabulary retrieval of spoken content with shorter/longer queries considering word/subword-based acoustic feature similarity," in *Interspeech*, 2012.

[45] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006 spoken term detection system," in *Interspeech*, 2007.

[46] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[47] P. Yu, K. Chen, L. Lu, and F. Seide, "Searching the audio notebook: keyword search in recorded conversations," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005.

[48] C. Chelba and A. Acero, "Position specific posterior lattices for indexing speech," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005.

[49] S. Parlak and M. Saraclar, "Spoken term detection for Turkish broadcast news," in *ICASSP*, 2008.

Hung-yi Lee, received the M.S. and Ph.D. degrees in communication engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2010 and 2012, respectively. From September 2012 to August 2013, he was a postdoctoral fellow in Research Center for Information Technology Innovation, Academia Sinica. He is currently visiting the Spoken Language Systems Group of MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). His research focuses on spoken content retrieval and spoken document summarization.

Sz-Rung Shiang, was born in 1990. She received the B.S. degrees in electrical engineering from National Taiwan University (NTU) in 2012. She is currently a Master student in Graduate Institute of Electrical Engineering, National Taiwan University, Taipei, Taiwan. Her research focused on automatic speech summarization.

Ching-feng Yeh, was born in 1987. He received the B.S. and M.S. degrees in electronic engineering and communication engineering from National Taiwan University (NTU), Taipei, Taiwan in 2009 and 2011, respectively.

He is currently pursuing the Ph.D. degree in the Department of Communication Engineering, National Taiwan University, Taipei, Taiwan. His research focused on automatic speech recognition.

Yun-Nung Chen, is a PhD student in the Language Technologies Institute of School of Computer Science at Carnegie Mellon University. Her research interests include spoken dialogue understanding, speech summarization, information extraction, and machine learning. She received Best Student Paper Awards from IEEE ASRU 2013 and IEEE SLT 2010, and a Best Student Paper Nominee from ISCA INTERSPEECH 2012. Chen has an MS in Language Technologies from Carnegie Mellon University, an MS and a BS in Computer Science and Information Engineering from National Taiwan University.

Yu Huang, was born in 1987. She received bachelor and master degrees in computer science and information engineering from National Taiwan University (NTU), Taipei, in 2009 and 2011, respectively.

Her research has been focused on semantic analysis, key term extraction and key term graph generation. Her paper "Automatic Key Term Extraction from Spoken Course Lectures Using Branching Entropy and Prosodic/Semantic Features" was awarded "Best Student Paper Award" in conference IEEE Spoken Language Technology, 2010.

Sheng-yi Kong, was born in 1981. He received the B.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University (NTU), Taipei, in 2004 and 2010, respectively.

From March 2010 to June 2010, he was an Research Intern mentored by Dr. Yao Qian under the supervision of Prof. Frank Soong with Microsoft Research Asia, Beijing, China. His research has been focused on spoken document summarization, spoken document clustering, semantic analysis, and information retrieval.

Lin-shan Lee, (F3) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

He has been a Professor of electrical engineering and computer science at the National Taiwan University, Taipei, Taiwan, since 1982 and holds a joint appointment as a Research Fellow of Academia Sinica, Taipei. His research interests include digital communications and spoken language processing. He developed several of the earliest versions of Chinese spoken language processing systems in the world including text-to-speech systems, natural language analyzers, dictation systems, and voice information retrieval systems.

Dr. Lee was Vice President for International Affairs (1996-1997) and the Awards Committee chair (1998-1999) of the IEEE Communications Society. He was a member of the Board of International Speech Communication Association (ISCA 2002-2009), a Distinguished Lecture (2007-2008) and a member of the Overview Paper Editorial Board (since 2009) of the IEEE Signal Processing Society, and the general chair of ICASSP 2009 in Taipei. He is a fellow of ISCA since 2010, and received the Meritorious Service Award from IEEE Signal Processing Society in 2011.