Segmentation of Lecture Videos Based on Text: A Method Combining Multiple Linguistic Features

Ming Lin
Department of Management Information, University of Arizona
mlin@cmi.arizona.edu

Jay F. Nunamaker Jr.
Department of Management Information, University of Arizona
nunamaker@cmi.arizona.edu

Michael Chau
School of Business, The University of Hong Kong
mchau@business.hku.hk

Hsinchun Chen
Department of Management Information Systems, University of Arizona
hchen@bpa.arizona.edu

Abstract

In multimedia-based e-Learning systems, there are strong needs for segmenting lecture videos into topic units in order to organize the videos for browsing and to provide search capability. Automatic segmentation is highly desired because of the high cost of manual segmentation. While a lot of research has been conducted on topic segmentation of transcribed spoken text, most attempts rely on domain-specific cues and formal presentation format, and require extensive training; none of these features exist in lecture videos with unscripted and spontaneous speech. In addition, lecture videos usually have few scene changes, which implies that the visual information that most video segmentation methods rely on is not available. Furthermore, even when there are scene changes, they do not match with the topic transitions. In this paper, we make use of the transcribed speech text extracted from the audio track of video to segment lecture videos into topics. We review related research and propose a new segmentation approach. Our approach utilizes features such as noun phrases and combines multiple content-based and discourse-based features. Our preliminary results show that the noun phrases are salient features and the combination of multiple features is promising to improve segmentation accuracy.

1. Introduction

Research has shown that multimedia instruction can enhance students' problem-solving skills [25, 33]. Recently, multimedia technology has become popular and been used extensively in e-Learning systems [2, 3]. Lectures are videotaped and used in e-Learning systems or Web-based systems [31]. The content of most lecture videos cover more than one topic or sub-topic. In order to facilitate student learning and minimize learning time, lecture videos usually are segmented into smaller topics for browsing. Content-based video retrieval also requires that video be divided into small pieces [31], because it is more useful to return short clips to a query instead of the whole video, as in the case with most video retrieval and information retrieval (IR) technologies. Since topic-based segmentation allows each segment to be a coherent topic, it also solves many problems stemming from the lack of context as in other non-topic based segmentation methods.

While manual video segmentation provides the highest quality, it is very time-consuming because an analyst has to watch the whole video several times in order to segment the video. Automatic segmentation is necessary and beneficial. In this paper, we define the segmentation task as a task of automatically segmenting videos into topically cohesive blocks by finding the boundaries where topics change. Video segmentation algorithms use various methods based on the input from multimedia streams, such as video, audio, and close caption. The most commonly used video segmentation methods rely on algorithms for shot-boundary (scene change) detection. Wactlar [28] used color histogram distance computation between successive images to detect scene changes. Zhang and Smoliar [32] proposed a method for progressive transition detection by combining both motion and statistical analysis. Although these segmentation

methods seem to be promising, the image cues that most of these methods rely on are not available for lecture videos. Lecture videos usually have very few scene changes (e.g. for many situations, there is only a "talking instructor" in the video) and in most cases those scene changes do not match with topic changes. On the other hand, the audio and the transcribed text extracted from videos provide rich content information for topic change detection. Thus our efforts in this paper are concentrated on topic segmentation using transcribed text. With the time stamps (extracted from automatic speech recognition software) that synchronize the video stream and transcribed text [5], it is possible to map the output of transcribed text segmentation back to video segmentation. Therefore, our video segmentation problem is transformed into the segmentation problem of transcribed spoken text.

Segmentation of transcribed spoken text also has been studied [1, 4]. Work in this area has been largely motivated by the TDT (Topic Detection and Tracking) initiative [1]. They usually focus on the broadcast and news domain in which the formal presentation format and cue phrases can be explored to improve segmentation accuracy. For instance, in CNN news stories, the phrase "This is Larry King…" normally implies the beginning or the ending of a new story or topic. In contrast, the speeches in lecture videos are typically unscripted and spontaneous Furthermore, a large set of training data, which is required for most of the methods used in TDT, are not available for lecture videos. Ultimately, the large variety of instructional styles of instructors in lectures makes the problem even more difficult.

Alternatively, without requiring formal presentation format and training, another area called "domain-independent text segmentation" provides possible methods to address this problem. Research in this area uses various content-based features such as word stem repetition [6, 8, 20], first use of words [23, 30], word frequency [23], and various knowledge sources such as WordNet and dictionaries [16] to segment written text based on lexical cohesion. In this paper we propose a method that combines multiple segmentation features to improve accuracy, which include noun phrases, topic noun phrases, verb classes, word stems, combined features, cue phrases, and pronouns.

The remainder of this paper is organized as follows. Section 2 reviews related research and identifies widely used segmentation features. Section 3 proposes our two-step approach which combines several features from literature. Section 4 describes an evaluation study, which compares our algorithm with a baseline approach and an existing algorithm. Finally, Section 5 concludes our research and outlines future directions.

2. Related Research

2.1. Text Segmentation

Most existing work in domain-independent text segmentation has been derived from Halliday and Hasan's lexical cohesion theory [8]. They proposed that text segments with similar vocabulary are likely to be in one coherent topic segment. Thus, finding topic boundaries could be done by detecting topic transitions from vocabulary. In this section, we review the literature using classifications based on different segmentation features, similarity measures, or methods of finding boundaries.

Researchers use different segmentation features to detect cohesion. Term repetition is dominant which includes different variants such as word stem repetition [9, 21, 30], word n-gram or phrases [12, 22], and word frequency [4, 23]. The first use of words also has been used [23, 30] because a large percentage of first-used words often accompanies topic shifts. Cohesion between semantically related words (e.g., synonyms, hyponyms, and collocational words) is captured using different knowledge sources such as thesaurus [18], dictionary [16], or large corpus [14, 19]. To measure the similarity between different text segments, research uses vector models [9], graphic methods [6, 21, 24], and statistical methods [27]. Methods for finding topic boundaries include sliding window [9], lexical chains [12, 16], dynamic programming [10, 19], and agglomerative clustering and divisive clustering [6, 29]. We describe some representative research with more details as follows. For a thorough review, please refer to [22].

Youmans [30] designed a technique based on the first uses of word types, called Vocabulary Management Profile. He pointed out that a large amount of first uses frequently followed topic boundaries. Kozima and Furugori [16] devised a measure called the Lexical Cohesion Profile (LCP) based on spreading activation within a semantic network derived from an English dictionary. The segment boundaries can be detected by the valleys (minimum values) of LCP. Hearst [9] developed a technique called TextTiling that automatically divides long expository texts into multi-paragraph segments using the vector space model, which has been widely used in information retrieval (IR). Topic boundaries are positioned where the similarity between neighboring blocks is low. Reynar [21]

described a method using optimization algorithm based on word repetition and a graphic technique called dotplotting. In [22], Reynar designed two algorithms for topic segmentation. The first is based solely on word frequency, represented by Katz's G-model [13]. The second one combines the first with other sources of evidence such as domain cues, content word bigram, and incorporates these features into a maximum entropy model. Choi [6]'s research was built on the work of Reynar [22]. The primary distinction is that inter-sentence similarity is replaced by rank in local context, and boundaries are discovered by divisive clustering.

2.2. Topic Segmentation in Lecture Context

Unlike the above segmentation methods that focus on written text, segmentation of transcribed spoken text is more challenging because spoken text lacks typographic cues such as headers, paragraphs, punctuation, and capitalized letters. Moreover, compared to written text and news stories, the topic boundaries within lecture transcripts tend to be more subtle and fuzzy because of the unscripted and spontaneous speech and the variety of instructional methods. Preliminary testing shows that the performance of one of the best text segmentation algorithms is even a little worse than that of baseline method (around 30%; refer to Section 4 for details). Therefore, we need more resolving power for segmenting lecture transcripts. In this paper, at first we propose that salient features such as noun phrases will improve segmentation accuracy because the name of concepts and theories that appear frequently in lectures are usually noun phrases. We also propose that combining multiple segment features to complement each other will lead to gains in resolving power and thus improve segmentation accuracy.

3. The Approach

Our approach utilizes the idea of sliding window similarly to TextTiling [9] and Kaufman [14] in terms of method of finding boundaries. We move a sliding window (e.g. 120 words) across the text by certain interval (e.g. 20 words). We compare the similarity between two neighboring windows (one gap), and then we draw a similarity graph for all the comparison or gaps (see Figure 1). The gap with lowest values (most dissimilar) are identified as possible topic boundaries.

As mentioned in the section above, the distinguished characteristic of our approach is that we use more salient features to gain resolving power and combine multiple features to complement each other. The core algorithm of approach has two steps:

- Preprocessing
- Finding boundaries

The preprocessing step is fairly standardized. Our algorithm takes the transcript text as input, and uses GATE [7] to handle tokenization, sentence splitting, and part-of-speech (POS) tagging. The POS tagger in GATE [11] is a modified version of the Brill tagger, which produces a part-of-speech tag as an annotation on each word or symbol. Porter's stemmer [20] was used for suffix stripping. Punctuations and uninformative words are removed using a stopword list.

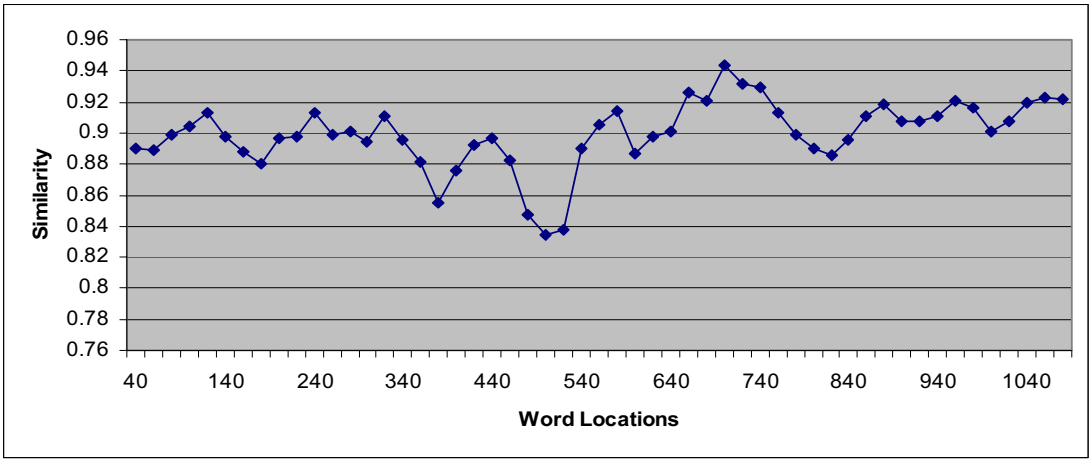

Figure 1. Example of a similarity graph

3.1. Feature Vectors

We identify the boundaries where the depth scores (differences between the similarly scores between two neighboring windows) are larger than a certain threshold. The crucial differences between our algorithm and TextTiling are the feature vectors used to represent the text window and the similarity measurement between two neighboring text windows.

We use seven feature vectors to represent each text window: noun phrases (NP), verb classes (VC), word stems (WS), topic words (TNP), combined features (NV), pronouns (PN), and cue phrases (CP). The first five features (NP, VC, WS, TNP and NV) are content-based features, which carry lexical or syntactic meanings. The last two features (PN and CP) are discourse-based features, which describe more about the properties of the text body surrounding the topic boundaries.

We use noun phrases instead of "bag of words" (single words) because noun phrases are usually more salient features and exhibit fewer sense ambiguities. Furthermore, most concepts are noun phrases. For example, in the transcript of a lecture video about search engines (see Figure 2), topic 13, "What's User Query" and topic 14, "Query Types" share a lot of words such as "query" and "keyword". The algorithms using "bag of words" features such as word repetition would not distinguish between these two topics. However, it will be much easier to separate these two topics if we use noun phrases ("query types" occurs several times in topic 14, but not in topic 13). We used the Arizona Noun Phraser [26] to extract the noun phrases from text.

Besides noun phrases, verbs also carry a lot of content information. Semantic verb classification has been used to characterize document type [15] because verbs typically embody an event's profile. Our intuition is that verb classification also represents topic information. After removing support verbs (e.g. is, have, get, go, etc., which do not carry a lot of content information), we use WordNet to build the links between verbs to provide a verb-based semantic profile for each text window during the segmentation process. WordNet is a lexical knowledge resource in which words are organized into synonym sets [17]. These synonym sets, or synsets, are connected by semantic relationships such as hypernymy or antonymy. We use the synonym and hypernymy relationship within two levels in WordNet. We only accept hypernymy relationships within two levels because of the flat nature of verb hierarchy in WordNet [15]. More concretely, when comparing two verbs between two text windows, they will be considered as having the same meaning if they are synonyms or hypernyms within fewer than two levels, or in other words in the same verb class. Except nouns and verbs, other content words such as adjectives and adverbs will be simply used in their stem forms.

****** 13. What's User Query***
After the indexing database is created for Web pages and searching strategy is implemented, a search engine is ready for searching.
When a user asks a ***query*** by typing a **keyword**, the search engine searches its database and finds all the Web documents that contain this **keyword**. Those documents are ranked based on their scored relevance to the ***query***. For example, if the ***query*** contains two **keywords** "news" and "weather," the CNN Web site will be retrieved because it contains both **keywords**.
****** 14. Query Types***
There are two primary ***query*** **types**.
One is **keyword** ***query***.
Basically it consists of a few **keywords** expressing user's information needs.
Users can use Boolean constraints to connect multiple **keywords**.
The commonly used Boolean constraints are AND, OR, and NOT.
For example, you can type a ***query*** [computer AND university].
The search engine will try to find the Web pages in which two **keywords** co-occur.
A user can also use double quotes to generate a phrase-query.
The other ***query*** type is natural language questions.
……

Figure 2. Part of the transcript for a lecture video about search engines

Other than those single features (nouns, verbs and word stems), we also have two complex features. The first one is topic terms, or, more exactly, topic noun phrases. Topic terms are defined as those terms with co-occurrence larger than one [13]. Topic terms usually hold more content information (such as "query type" in Figure 1), which means they should carry more weight in our algorithm. The other complex feature is a combined feature of nouns and verbs. We extract the main noun and verb in each sentence according to the POS tags, with the expectation of capturing the complex relationship information of subject plus behavior.

Different from the above five content-based features, the two discourse-based features focus on the small size text body surrounding the pseudo-boundaries proposed by the algorithm based on the five content-based features. We use a size of five words in our algorithm. In other words, we check the five words before and after the pseudo-boundaries. If we find any pronoun (from a pronoun list) within the five-word window, we decrease the possibility score of this pseudo-boundary as a true boundary. The reason is that pronouns usually substitute for nouns or noun phrases that appear within the same topic. Any occurrence of cue phrases (from a cue phrase list) will increase the possibility of pseudo-boundary as a true boundary because cue phrases usually indicate the change of discourse structure. We use the general cue phrases list (Table 1) and the pronoun list (Table 2) from [22].

Table 1. Cue phrases

actually	further	otherwise
also	furthermore	right
although	generally	say
and	however	second
basically	indeed	see
because	like	similarly
but	look	since
essentially	next	so
except	no	then
finally	now	therefore
first	ok	well
firstly	or	yes

Table 2. Pronouns

she
her
hers
herself
he
him
his
himself
they
their
them
theirs
themselves

3.2. Similarity Measure

The similarity between two neighboring text windows (w_1 and w_2) is calculated by the cosine measure. Given two neighboring text windows, their similarity score is the weighted sum of the cosine products of seven feature vectors.

$$Similarity(w_1, w_2) = \sum_j \frac{\sum_i f_{j,i,w1} f_{j,i,w2}}{\sqrt{\sum_i f^2_{j,i,w1} \sum_i f^2_{j,i,w2}}} S_j$$

j represents the different features (1 to 7 here), and i ranges over all the specific feature weight values (e.g. noun phrases) in the text window. $f_{j,i,w1}$ is the i-th feature weight value of j-th type feature vector in text window w_1. We calculate $f_{j,i,w1}$ based on a formula similar to the *TF*IDF* formula which is widely used in information retrieval literature. We call our measure *TF*ISF*. *TF* is the term frequency and is represented by the repetition times of terms (e.g. noun phrases) within a text window. We adapt the concept of Inversed Document Frequency (*IDF*) as Inversed Segment Frequency (*ISF*): $ISF = log(N/n)$. N is the number of text windows in the text transcript, and n is the number of text windows in which feature (j,i) occurs at least once. j is the feature type and i is the specific word or noun phrase in the feature vector. S_j is the significant value of some specific feature type. The best way to calculate S_j is to use language model or word model and utilize large

corpus. For example, Reynar [23] uses G-model and Wall Street Journal to calculate S_j (called word frequency in [23]). At current stage, without appropriate large training corpus, the significant values S_j are calculate based on human heuristics and hand tuning. We assume that significance of the five features are in the following order: S(TNP) > S(NV) > S(NP) > S(VC) > S(WS).

4. Evaluation

To validate our proposal that noun phrases are salient features and that the combination of features improve accuracy, we choose a subset of four features (NP, TNP, CP, PN) from the seven proposed to conduct a preliminary experiment. We evaluate our algorithm, called PowerSeg (with the subset of features), by comparing its performance to that of a baseline method, and TextTiling [9], one of the best text segmentation algorithms. We use the Java version implementation of TextTiling from Choi [6]. We also have developed a simple version of the Baseline segmentation algorithm. Given the average number of segments of the whole data set as prior knowledge, the baseline algorithm randomly chooses a point (some sentence number) to be a boundary.

4.1. Data Set and Performance Metrics

Since there is no available annotated corpus for lectures, we use the lecture transcripts in our e-Learning system called Learning By Asking (LBA) [31] as pilot data for evaluation. Due to the limited number of transcripts in LBA, we choose a small data set of three transcripts for our preliminary experiment. One transcript is from a lecture about the Internet and search engines, and the other two transcripts are from a database course. The average length of videos is around 28 minutes and the average number of words in the transcripts is 1859. All three transcripts are segmented by experts (the original instructors). We assume the segmentation results from experts are perfect (100% accuracy). The performance measures of PowerSeg, TextTiling, and Baseline are calculated by comparing their output results to the results from experts.

Selecting an appropriate performance measure for our purpose is difficult. Metrics suggested by [4] is well accepted and has been adopted by TDT. It measures the probability that two sentences drawn at random from a corpus are correctly classified as to whether they belong to the same story. However, this metrics cannot fulfill our purpose because it requires some knowledge of the whole large collection and it also is not clear how to combine the scores from probabilistic metrics when segmenting collections of texts in different files [22]. Finally, we chose precision, recall and F-measure as our metrics. We chose precision and recall because they are well accepted and frequently used in information retrieval and text segmentation literature [9, 22]. F-measure was chosen to overcome the tuning effects of precision and recall. They are defined as follows:

$$\mathrm{P(recision)} = \frac{\mathrm{No\_of\_Matched\_Boundaries}}{\mathrm{No\_of\_Hypothesized\_Boundaries}}$$

$$\mathrm{R(ecall)} = \frac{\mathrm{No\_of\_Matched\_Boundaries}}{\mathrm{No\_of\_Actual\_Boundaries}}$$

$$\mathrm{F\text{-}Measure} = \frac{2PR}{P+R}$$

No_of_Matched_Boundaries is the number of correctly identified or matched boundaries when comparing to actual boundaries identified by experts. *No_of_Hypothesized_Boundaries* is the number of boundaries proposed by the algorithm (e.g. PowerSeg). Besides exact match, we also used the concept of fuzzy boundary which implies that hypothesized boundaries a few sentences (usually 1) away from the actual boundaries are also considered as correct. We used fuzzy boundary because for most lengthy lecture videos, one sentence away from the actual boundary is only a very short time period when we map the transcript back to the video, which is acceptable for general learning purpose. For instance, for our data set the average time span of one sentence is 0.2 minutes, or 12 seconds.

4.2. Experiment and Results

We ran the three algorithms (Baseline, TextTiling and PowerSeg) using the three transcripts and calculated the mean performance. We measured the performance using precision, recall and F-Measure. We also calculated the performance measures under both conditions: exact match and fuzzy boundary (allowing a hypothesized boundary to be one sentence away from the true boundary).

At first, in order to test whether noun phrase (NP) are salient features, we ran a PowerSeg version with NP feature only. We found that even with NP only, PowerSeg improve the performance (F-Measure) by more than 10% comparing to Baseline and TextTiling for both “exact match” and “fuzzy boundary” conditions (Table 3). When we used “fuzzy boundary”, the performance increased dramatically (around 23% for PowerSeg) as we expected.

Table 3. Comparison of algorithms

Algorithms	Exact Match			Fuzzy (1)		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Baseline	0.32	0.32	0.32	0.56	0.56	0.56
TextTiling	0.30	0.18	0.22	0.75	0.46	0.56
PowerSeg (NP)	0.41	0.35	0.37	0.77	0.67	0.70

However, because two of the three transcripts have very small segments (3-5 sentences), fuzzy boundary (one sentence away from the actual boundary) makes the algorithms easy to perform well.

In order to evaluate the effectiveness of feature combination, we ran four different versions of PowerSeg which used 4 types of feature subsets: WS (word stem), NP (noun phrase), NP+TNP (noun phrase plus topic noun phrases) and NP+CP+PN (noun phrases, cue phrases, and pronouns) (Table 4). We found that the combination of noun phrases, cue phrases, and pronouns has a better performance than using noun phrases only (NP). This confirms our original hypothesis that the combination of multiple features, especially combination of content-based features and discourse-based features, improve segmentation accuracy (F-Measure). However, the improvement is relatively small, only around 2%. The possible reason is that the cue phrase list and pronoun list we used is too general, and our data set is small. To our surprise, the NP+TNP combination

Table 4. Comparison of PowerSeg with different feature subset

Features Combination	Exact Match			Fuzzy (1)		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Baseline	0.32	0.32	0.32	0.56	0.56	0.56
WS	0.30	0.18	0.22	0.75	0.46	0.56
NP	0.41	0.35	0.37	0.77	0.67	0.70
NP+TNP	0.39	0.32	0.34	0.73	0.60	0.65
NP+CP+PN	0.42	0.37	0.39	0.77	0.68	0.72

performed slightly worse than using NP only. The first possible reason is that although we define topic noun phrases as those noun phrases with frequency larger than 1, our feature weight and calculation of similarity are still based on term frequencies (refer to section 3.2.). When we calculate the similarity between two text windows, TNPs already occupy a large percentage of weight. From another perspective, it also shows that the complementary features such as content-based features and discourse-based features will improve performance, not those with similar characteristics such as noun phrases and topic noun phrases.

We also tested the effects of algorithm parameters using one sample transcript (the one about the Internet and search engines). PowerSeg has two parameters: w and s. w is the size of the sliding text window in terms of words, and s is the step size that the text window slides each time. The experiment results (Table 5) showed that the algorithm performed best when the size of text window (w = 120) approximates the size of actual segment (the actual average segment size of this transcript is 171 words). Further, the experiment results (Table 6) also showed that relatively smaller step size (s = 20) produced more sensible output (F-Measure is the highest: 0.65).

Table 5. Effect of sliding window size

No.	Parameters (*w*, *s*)	Precision	Recall	F-Measure
1	(60, 10)	0.33	0.79	0.47
2	(120, 20)	0.58	0.74	0.65
3	(240, 40)	0.55	0.32	0.40

Table 6. Effect of step size

No.	Parameters (*w*, *s*)	Precision	Recall	F-Measure
1	(120, 20)	0.58	0.74	0.65
2	(120, 60)	0.50	0.21	0.30
3	(120, 120)	0.80	0.21	0.33

5. Conclusion and Future Directions

With the purpose of segmenting lecture videos with unscripted and spontaneous speech, we proposed a video segmentation approach based on transcribed text. Our approach utilized salient segmentation features and combined content-based and discourse-based features to gain more resolving power. Our preliminary experiment results demonstrated that the effectiveness of noun phrases as salient features and the methodology of combining multiple segmentation features to complement each other is promising. One of our future directions is to implement the full algorithm incorporating all proposed features, and test the effectiveness of the combination of different features to find the set of most salient segmentation features. One of the weaknesses of our algorithm is that we had to hand-tune the parameters, which is not very efficient. To address this problem, we plan to use machine-learning methods such as decision tree instead of hand-tuning in our future research.

6. References

[1] Allan, J., Carbonell, J., Doddington, G., Yamron, J. and Yang, Y. "Topic detection and tracking pilot study: Final report," In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[2] Agius, H. W., and Angelides, M. C. "Developing knowledge-based intelligent multimedia tutoring systems using semantic content-based modeling," *Artificial Intelligence Review*, 1999, 13, pp. 55-83.

[3] Baltes, C. "The E-learning balancing act: training and education with multimedia," *IEEE Multimedia*, 2001. 8(4). pp. 16-19.

[4] Beeferman, D., Berger, A. and Lafferty, J. "Text segmentation using exponential models," In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 1997, pp. 35-46.

[5] Blei, D. M. and Moreno, P. J. "Topic segmentation with an aspect hidden Markov model," *Proceedings of the 24th International Conference on Research and Development in Information Retrieval (SIGIR 2001),* New York, NY: ACM Press, 2001

[6] Choi, F. "Advances in domain independent linear text segmentation," *The North American Chapter of the Association for Computational Linguistics (NAACL)*, Seattle, USA, 2000.

[7] Cunningham, H. *Software Architecture for Language Engineering*. PhD Thesis, University of Sheffield, 2000.

[8] Halliday, M. and Hasan, R. *Cohesion in English*, Longman 1976

[9] Hearst, M. A. "TextTiling: Segmenting Text intoMulti-paragraph Subtopic Passages," *Computational Linguistics*, vol 23(1), pp. 33-64, 1994.

[10] Heinonen, O. "Optimal Multi-Paragraph Text Segmentation by Dynamic Programming," In *Proceedings of 17th International Conference on Computational Linguistics (COLING-A CL98)*, 1998. pp.1484-1486.

[11] Hepple, M. "Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers," In *Proceedings of the 38th Annual Meeting*

of the Association for Computational Linguistics (ACL-2000), Hong Kong, October 2000.

[12] Kan, M., Klavans, J.L. and McKeown, K. R. "Linear segmentation and segment significance," In *Proceedings of the 6th International Workshop of Very Large Corpora*, 1998. pp. 197-205.

[13] Katz, S. M. "Distribution of content words and phrases in text and language modeling," *Natural Language Engineering*, 1996. 2(1):15-59.

[14] Kaufmann, S. "Cohesion and collocation: Using context vectors in text segmentation," In Proceedings *of the 37th Annual Meeting of the Association of for computational Linguistics* (Student Session), 1999. pp. 591-595, College Park, USA, June. ACL.

[15] Klavans, J. and Kan, M.Y. "Role of verbs in document analysis," In *COLING-ACL*, 1998. pp. 680-686.

[16] Kozima, H. and Furugori, T. "Similarity between words computed by spreading activation on an English dictionary," In *Proceedings of the European Association for Computational Linguistics*, 1993, pp. 232--239.

[17] Miller, G., Beckwith, R., Felbaum, C., Gross, D., and Miller, K. "Introduction to WordNet: An online lexical database," *International Journal of Lexicography* (special issue), 1990. 3(4). pp. 235-312.

[18] Morris, J. and Hirst, G. "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," *Computational Linguistics*, 1991. (17). pp. 21-48.

[19] Ponte, J. M. and Croft, W. B. "Text segmentation by topic," In *European Conference on Digital Libraries*, 1997. pp. 113-125, Pisa, Italy.

[20] Porter, M. "An algorithm for suffix stripping," *Program*, 1980. 14(3). pp. 130-137.

[21] Reynar, J. C. "An automatic method of finding topic boundaries", In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Student Session, pp. 331-333, Las Cruces, New Mexico, 1994.

[22] Reynar, J. C. *Topic segmentation: Algorithms and applications*, PhD thesis, Computer and Information Science, University of Pennsylvania, 1998.

[23] Reynar, J. C. "Statistical models for topic segmentation," In *Proceedings of 37th Annual Meeting of the ACL*, 1999. pp. 357-364.

[24] Salton, G., Singhal, A., Buckley, C. and Mitra, M. "Automatic text decomposition using text segments and text themes," In *Proceedings of Hypertext'96*, 1996. ACM Press, New York, pp. 53-65.

[25] Sean, J. A. "Capitalising on Interactive Multimedia Technologies in Dynamic Environments," 1997. Available: http://crm.hct.ac.ae/021senn.html.

[26] Tolle, K. and Chen, H. "Comparing Noun Phrasing Techniques for Use with Medical Digital Library Tools," *Journal of the American Society for Information Science, Special Issue on Digital Libraries*, 2000 51(4). pp. 352-370.

[27] Utiyama, M. and Isahara, H. "A statistical model for domain - independent text segmentation," In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, 2001. pp.491-498.

[28] Wactlar, H. D. "Informedia - search and summarization in the video medium," Imagina 2000 Conference, Monaco, 2000.

[29] Yaari, Y. "Segmentation of expository texts by hierarchical agglomerative clustering," In Proceedings of Recent Advances in Natural Language Processing, Bulgaria, 1997.

[30] Youmans, G. "A new tool for discourse analysis: The vocabulary management profile," Language. 1991. 67(4). pp. 763--789.

[31] Zhang, D. S. Virtual mentor and media structuralization theory, PhD thesis, University of Arizona, Tucson, AZ, 2002.

[32] Zhang, H. J. and Smoliar, S. W. "Developing power tools for video indexing and retrieval," SPIE'94 Storage and Retrieval for Video Databases, San Jose, CA, USA, 1994.

[33] Zhang, W. "Multimedia, Technology, Education and Learning," Technological Innovations in Literacy and Social Studies Education, 1995. The University of Missouri-Columbia.