# Imposing Hierarchical Browsing Structures onto Spoken Documents

**Xiaodan Zhu & Colin Cherry**
Institute for Information Technology
National Research Council Canada
{Xiaodan.Zhu,Colin.Cherry}@nrc-cnrc.gc.ca

**Gerald Penn**
Department of Computer Science
University of Toronto
gpenn@cs.toronto.edu

## Abstract

This paper studies the problem of imposing a known hierarchical structure onto an unstructured spoken document, aiming to help browse such archives. We formulate our solutions within a dynamic-programming-based alignment framework and use minimum error-rate training to combine a number of global and hierarchical constraints. This pragmatic approach is computationally efficient. Results show that it outperforms a baseline that ignores the hierarchical and global features and the improvement is consistent on transcripts with different WERs. Directly imposing such hierarchical structures onto raw speech without using transcripts yields competitive results.

## 1 Introduction

Though speech has long served as a basic method of human communication, revisiting and browsing speech content had never been a possibility before human can record their own voice. Recent technological advances in recording, compressing, and distributing such archives have led to the consistently increasing availability of spoken content.

Along with this availability comes a demand for better ways to browse such archives, which is inherently more difficult than browsing text. In relying on human beings' ability to browse text, a solution is therefore to reduce the speech browsing problem to a text browsing task through technologies that can automatically convert speech to text, i.e., the automatic speech recognition (ASR). Research along this line has implicitly changed the traditional speaking-for-hearing and writing-for-reading construals: now speech can be *read* through its transcripts, though it was not originally intended for this purpose, which in turn raises a new set of problems.

The efficiency and convenience of reading spoken documents are affected by at least two facts. First, the quality of transcripts can impair browsing efficiency, e.g., as shown in (Stark et al., 2000; Munteanu et al., 2006), though if the goal is only to browse salient excerpts, recognition errors on the extracts can be reduced by considering the confidence scores assigned by ASR (Zechner and Waibel, 2000; Hori and Furui, 2003).

Even if transcription quality is not a problem, browsing transcripts is not straightforward. When intended to be read, written documents are almost always presented as more than uninterrupted strings of text. Consider that for many written documents, e.g., books, indicative structures such as section/subsection headings and tables-of-contents are standard constituents created manually to help readers. Structures of this kind, however, are rarely aligned with spoken documents.

In this paper, we are interested in addressing the second issue: adding hierarchical browsable structures to speech transcripts. We define a hierarchical browsable structure as a set of nested labelled bracketing which, when placed in text, partition the document into labeled segments. Examples include the sequence of numbered section headings in this paper, or the hierarchical slide/bullet structure in the slides of a presentation.

*Coling 2010: Poster Volume*, pages 1550–1557,
Beijing, August 2010

An ideal solution to this task would directly infer both the hierarchical structure and the labels from unstructured spoken documents. However, this is a very complex task, involving the analysis of not only local but also high-level discourse over large spans of transcribed speech. Specifically for spoken documents, spoken-language characteristics as well as the lack of formality and thematic boundaries in transcripts violate many conditions that a reliable algorithm (Marcu, 2000) relies on and therefore make the task even harder.

In this paper, we aim at a less ambitious but naturally occurring problem: imposing a known hierarchical structure, e.g., presentation slides, onto the corresponding document, e.g., presentation transcripts. Given an ordered, nested set of topic labels, we must place the labels so as to correctly segment the document into appropriate units. Such an alignment would provide a useful tool for presentation browsing, where a user could easily navigate through a presentation by clicking on bullets in the presentation slides. The solution to this task should also provide insights and techniques that will be useful in the harder structure-inference task, where hierarchies and labels are not given.

We present a dynamic-programming-based alignment framework that considers global document features and local hierarchical features. This pragmatic approach is computationally efficient and outperforms a baseline alignment that ignores the hierarchical structure of bullets within slides. We also explore the impact of speech recognition errors on this task. Furthermore, we study the feasibility of directly aligning a structure to raw speech, as opposed to a transcript.

## 2 Related work

**Topic/slide boundary detection** The previous work most directly related to ours is research that attempts to find *flat* structures of spoken documents, such as topic and slide boundaries. For example, the work of (Chen and Heng, 2003; Ruddarraju, 2006; Zhu et al., 2008) aims to find slide boundaries in the corresponding lecture transcripts. Malioutov et al. (2007) developed an approach to detecting topic boundaries of lecture recordings by finding repeated acoustic patterns. None of this work, however, has involved hierarchical structures that exist at different levels of a document.

In addition, researchers have also analyzed other multimedia channels, e.g., video (Liu et al., 2002; Wang et al., 2003; Fan et al., 2006), to detect slide transitions. Such approaches, however, are unlikely to find semantic structures that are more detailed than slide transitions, e.g., the bullet hierarchical structures that we are interested in.

**Building tables-of-contents on written text** A notable effort going further than topic segmentation is the work by Branavan et al. (2007), which aims at the ultimate goal of building tables-of-contents for written texts. However, the authors assumed the availability of the hierarchical structures and the corresponding text spans. Therefore, their problem was restricted to generating titles for each span. Our work here can be thought of as the inverse problem, in which the title of each section is known, but the corresponding segments in the spoken documents are unknown. Once the correspondence is found, an existing hierarchical structure along with its indicative titles is automatically imposed on the speech recordings. Moreover, this paper studies spoken documents instead of written text. We believe it is more attractive not only because of the necessity of browsing spoken content in a more efficient way but also the general absence of helpful browsing structures that are often available in written text, as we have already discussed above.

**Rhetoric analysis** In general, analyzing discourse structures can provide thematic skeletons (often represented as trees) of a document as well as relationship between the nodes in the trees. Examples include the widely known discourse parsing work by Marcu (2000). However, when the task involves the understanding of high-level discourse, it becomes more challenging than just finding local discourse conveyed on small spans of text; e.g., the latter is more likely to benefit from the presence of discourse markers. Specifically for spoken documents, spoken-language characteristics as well as the absence of formality and thematic boundaries in transcripts pose additional difficulty. For example, the boundaries of sentences, paragraphs, and larger text blocks like sections are often missing. Together with speech recognition errors as well as other speech characteristics such as speech disfluences, they will impair the conditions on which an effective and reliable algorithm of discourse analysis is often built.

## 3 Problem formulation

We are given a speech sequence $U = u_1, u_2, ..., u_m$, where $u_i$ is an utterance. Depending on the application, $u_i$ can either stand for the audio or transcript of the utterance. We are also given a corresponding hierarchical structure. In our work, this is a sequence of lecture slides containing a set of slide titles and bullets, $B = \{b_1, b_2, ..., b_n\}$, organized in a tree structure $T(\Re, \aleph, \Psi)$, where $\Re$ is the root of the tree that concatenates all slides of a lecture; i.e., each slide is a child of the root $\Re$ and each slide's bullets form a subtree. In the rest of this paper, the word *bullet* means both the title of a slide (if any) and any bullet in it. $\aleph$ is the set of nodes of the tree (both terminal and non-terminals, excluding the root $\Re$), each corresponding to a bullet $b_i$ in the slides. $\Psi$ is the edge set. With the definitions, our task is herein to find the triple $(b_i, u_k, u_l)$, denoting that a bullet $b_i$ starts from the $kth$ utterance $u_k$ and ends at the $lth$. Constrained by the tree structure, the text span corresponding to an ancestor bullet contains those corresponding to its descendants; i.e., if a bullet $b_i$ is the ancestor of another bullet $b_j$ in the tree, the acquired boundary triples $(b_i, u_{k1}, u_{l1})$ and $(b_j, u_{k2}, u_{l2})$ should satisfy $u_{k1} \leq u_{k2}$ and $u_{l1} \geq u_{l2}$. In implementation, we only need to find the starting point of a bullet, i.e., a pair $(b_i, u_k)$, since we know the tree structure in advance and therefore we know that the starting position of the next sibling bullet is the ending boundary for the current bullet.

## 4 Our approaches

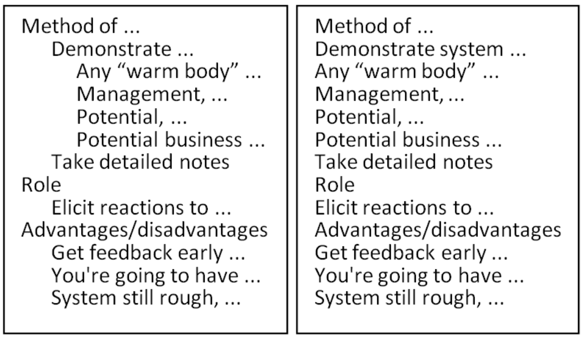
Our task is to find the correspondence between slide bullets and a speech sequence or its transcripts. Research on finding correspondences between parallel texts pervades natural language processing. For example, aligning bilingual sentence pairs is an essential step in training machine translation models. In text summarization, the correspondence between human-written summaries and their original texts has been identified (Jing, 2002), too. In speech recognition, forced alignment is applied to align speech and transcripts. In this paper, we keep the general framework of alignment in solving our problem.

Our solution, however, should be flexible to consider multiple constraints such as those conveyed in hierarchical bullet structures and global word distribution. Accordingly, the model proposed in this paper depends on two orthogonal strategies to ensure efficiency and richness of the model. First of all, we formulate all our solutions within a classic dynamic programming framework to enforce computational efficiency (section 4.1). On the other hand, we explore the approach to incorporating hierarchical and global features into the alignment framework (Section 4.2). The associated parameters are then optimized with Powell's algorithm (Section 4.3).

### 4.1 A pre-order walk of bullet trees

We formulate our solutions within the classic dynamic-programming-based alignment framework, dynamic time warping (DTW). To this end, we need to sequentialize the given hierarchies, i.e., bullet trees. We propose to do so through a pre-order walk of a bullet tree; i.e., at any step of a recursive traversal of the tree, the alignment model always visits the root first, followed by its children in a left-to-right order. This sequentialization actually corresponds to a reasonable assumption: words appearing earlier on a given slide are spoken earlier by the speaker. The pre-order walk is also used by (Branavan et al., 2007) to reduce the search space of their discriminative table-of-contents generation. Our sequentialization strategy can be intuitively thought of as removing indentations that lead each bullet. As shown in Figure 1, the right panel is a bullet array resulting from a pre-walk of the slide in the left panel. In our baseline model, the resulted bullet array is directly aligned with lecture utterances.

Other orders of bullet traversal could also be considered, e.g., when speech does not strictly follow bullet orders. In general, one can regard our task here as a tagging problem to allow further flexibility on bullet-utterance correspondence, in which bullets are thought of as tags. However, considering the fact that bullets are created to organize speech and in most cases they correspond to the development of speech content monotonically, this paper focuses on addressing the problem in the alignment framework.



```
Method of ...
    Demonstrate ...
        Any "warm body" ...
        Management, ...
        Potential, ...
        Potential business ...
    Take detailed notes
Role
    Elicit reactions to ...
Advantages/disadvantages
    Get feedback early ...
    You're going to have ...
    System still rough, ...
```

```
Method of ...
Demonstrate system ...
Any "warm body" ...
Management, ...
Potential, ...
Potential business ...
Take detailed notes
Role
Elicit reactions to ...
Advantages/disadvantages
Get feedback early ...
You're going to have ...
System still rough, ...
```

Figure 1: A pre-order walk of a bullet tree.

### 4.2 Incorporating hierarchical and global features

Our models should be flexible enough to consider constraints that could be helpful, e.g., the hierarchical bullet structures and global word distribution. We propose to consider all these constraints in the phase of estimating similarity matrices. To this end, we use two levels of similarity matrices to capture local tree constraints and global word distributions, respectively.

First of all, information conveyed in the hierarchies of bullet trees should be considered, such as the potentially discriminative nature between two sibling bullets (Branavan et al., 2007) and the relationships between ancestor and descendant bullets. We incorporate them in the bullet-utterance similarity matrices. Specifically, when estimating the similarity between a bullet $b_i$ and an utterance $u_j$, we consider local tree constraints based on where the node $b_i$ is located on the slide. We do so by accounting for first and second-order tree features. Given a bullet, $b_i$, we first represent it as multiple vectors, one for each of the following: its own words, the words appearing in its parent bullet, grandparent, children, grandchildren, and the bullets immediately adjacent to $b_i$. That is, $b_i$ is now represented as 6 vectors of words (we do not discriminate between its left and right siblings and put these words in the same vector). Similarity between the bullet $b_i$ and an utterance $u_j$ is calculated by taking a weighted average over the similarities between each of the 6 vectors and the utterance $u_j$. A linear combination is used and the weights are optimized on a development set.

Global property of word distributions could be helpful, too. A general term often has less discriminative power in the alignment framework than a word that is localized to a subsection of the document and is related to specific subtopics. For example, in a lecture that teaches introductory computer science topics, aligning a general term "computer" should receive a smaller weight than aligning some topic-specific terms such as "automaton." The latter word is more likely to appear in a more narrow text span. It is not straightforward to directly calculate *idf* scores unless a lecture is split into smaller segments in some way. Instead, in our models, the distribution property of a word is considered in word-level similarity matrices with the following formula.

$$sim(w_i, w_j) = \begin{cases} 0 & : i \neq j \\ 1 - \lambda \, \frac{var(w_i)}{\max_k(var(w_k))} & : i = j \end{cases}$$

Aligning different words receives no bonus, while matching the same word between bullets and utterances receives a score of 1 minus a distribution penalty, as shown in the formula above. The function $var(w_i)$ calculates the standard variance of the positions where the word $w_i$ appears. Divided by the maximal standard variance of word positions in the same lecture, the score is normalized to [0,1]. This distribution penalty is weighted by $\lambda$, which is tuned in a development set. Again, a general term is expected to have a larger positional variance.

Once a word-level matrix is acquired, it is combined with the bullet-utterance level matrix discussed above. Specifically, when measuring the similarity between a word vector (one of the 6 vectors) and the transcripts of an utterance, we sum up the word-level similarity scores of all matching words between them, normalize the resulted score by the length of the vector and utterance, and then renormalize it to the range

[0, 1] within the same spoken document. The final bullet-utterance similarity matrix is incorporated into the pre-order-walk suquentialization discussed above, when alignment is conducted.

### 4.3 Parameter optimization

Powell's algorithm (Press et al., 2007) is used to find the optimal weights for the constraints we incorporated above, to directly minimize the objective function, i.e., the $P_k$ and WindowDiff scores that we will discuss later. As a summary, we have 7 weights to tune: a weight for each of the following: parent bullet, grandparent, adjacent siblings, children, grandchildren, and the current bullet, plus the word distribution penalty $\lambda$. The values of these weights are determined on a development set.

Note that the model we propose here does not exclude the use of further features; instead, many other features, such as smoothed word similarity scores, can be easily added to this model. We are conservative on our model complexity here, in terms of number of weights need to be tuned, for the consideration of the size of data that we can used to estimate these weights. Finally, with all the 7 weights being determined, we apply the standard dynamic time warping (DTW).

## 5 Experimental set-up

### 5.1 Data

We use a corpus of lectures recorded at a large research university. The correspondence between bullets and speech utterances are manually annotated in a subset of this lecture corpus, which contains approximately 30,000 word tokens in its manual transcripts. Intuitively, this roughly equals a 120-page double-spaced essay in length. The lecturer's voice was recorded with a head-mounted microphone with a 16kHz sampling rate and 16-bit samples. Students' comments and questions were not recorded. The speech is split into utterances by pauses longer than 200ms, resulting in around 4000 utterances. There are 119 slides that are composed of 921 bullets. A subset containing around 25% consecutive slides and their corresponding speech/transcripts are used as our development set to tune the parameters discussed earlier; the rest data are used as our test set.

### 5.2 Evaluation metric

We evaluate our systems according to how well the segmentation implied by the inferred bullet alignment matches that of the manually annotated gold-standard bullet alignment. Though one may consider that different bullets may be of different importance, in this paper we do not use any heuristics to judge this and we treat all bullets equally in our evaluation. We evaluate our systems with the $P_k$ and WindowDiff metrics (Malioutov et al., 2007; Beeferman et al., 1999; Pevsner and Hearst, 2002). Note that for both metrics, the lower a score is, the better the performance of a system is. The $P_k$ score computes the probability of a randomly chosen pair of words being inconsistently separated. The WindowDiff is a variant of $P_k$; it penalizes false positives and near misses equally.

## 6 Experimental results

### 6.1 Alignment performance

Table 1 presents the results on automatic transcripts with a 39% WER, a typical WER in realistic and uncontrolled lecture conditions (Leeuwis et al., 2003; Hsu and Glass, 2006). The transcripts were generated with the SONIC toolkit (Pellom, 2001). The acoustic model was trained on the Wall Street Journal dictation corpus. The language model was trained on corpora obtained from the Web through searching the words appearing on slides as suggested by (Munteanu et al., 2007).

| | $Pk$ | WindowDiff |
|---|---|---|
| UNI | 0.481 | 0.545 |
| TT | 0.469 | 0.534 |
| B-ALN | 0.283 | 0.376 |
| HG-ALN | 0.266 | 0.359 |

Table 1: The $P_k$ and WindowDiff scores of uniform segmentation (UNI), TextTiling (TT), baseline alignment (B-ALN), and alignment with hierarchical and global information (HG-ALN).

From Table 1, we can see that the model that utilizes the hierarchical structures of slides and global distribution of words, i.e., the HG-ALN model, reduces both $P_k$ and WindowDiff scores over the baseline model, B-ALN. As discussed earlier, the baseline is a re-implementation of standard dynamic time warping based only on a pre-order walk of the slides, while the HG-ALN model incorporates also hierarchical bullet constraints and global word distribution.

Table 1 also presents the performance of a typical topic segmentation algorithm, TextTiling (Hearst, 1997). Note that similar to (Malioutov et al., 2007), we force the number of predicted topic segments to be the target number, i.e., in our task, the number of bullets. The results show that both the $P_k$ and WindowDiff scores of TextTiling are significantly higher than those of the alignment algorithms. Our manual analysis suggests that many segments are as short as several utterances and the difference between two consecutive segments is too subtle to be captured by a lexical cohesion-based method such as TextTiling. For comparison, We also present the results of uniform segmentation (UNI), which simply splits the transcript of each lecture evenly into segments with same numbers of words.

### 6.2 Performance under different WERs

Speech recognition errors within reasonable ranges often have very small impact on many spoken language processing tasks such as spoken language retrieval (Garofolo et al., 2000) and speech summarization (Christensen et al., 2004; Maskey, 2008; Murray, 2008; Zhu, 2010). To study the impact of speech recognition errors on our task here, we experimented with the alignment models on manual transcripts as well as on automatic transcripts with different WERs, including a 39% and a 46% WER produced by two real recognition systems. To increase the spectrum of our observation, we also overfit our ASR models to obtain smaller WERs at the levels of 11%, 19%, and 30%.

From Figure 2, we can see that at all levels of these different WERs, the HG-ALN model consistently outperforms the B-ALN system (the AUDIO model will be discussed below). The $P_k$ and WindowDiff curves also show that the alignment performance is sensitive to recognition errors, particularly when the WER is in the range of 30%–45%, suggesting that the problem we study here can benefit from the improvement of current ASR systems in this range, e.g., the recent advance achieved in (Glass et al., 2007).

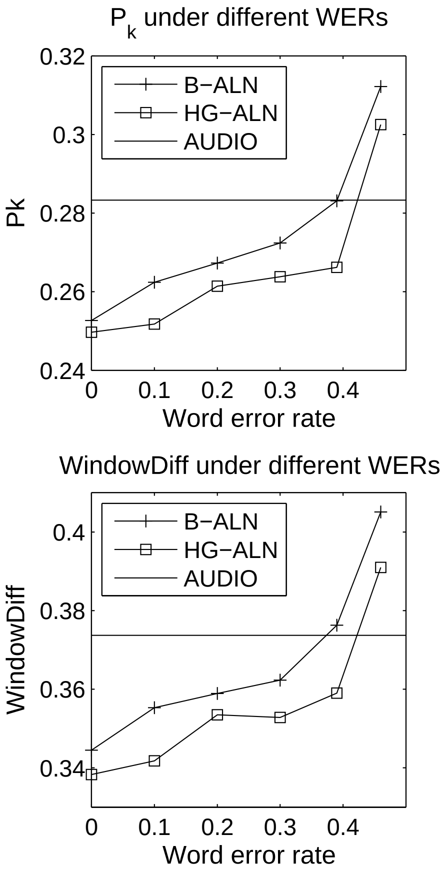


Figure 2: The impact of different WERs on the alignment models. The performance of an audio-based model (AUDIO) is also presented.

### 6.3 Imposing hierarchical structures onto raw speech

We can actually impose hierarchical structures directly onto raw speech, through estimating the similarity between bullets and speech. This enables navigation through the raw speech by using slides; e.g., one can hear different parts of speech by clicking a bullet. We apply keyword spotting to solve this problem, which detects the occurrences of each bullet word in the corresponding lecture audio.

In this paper, we use a token-passing based algorithm provided in the ASR toolkit SONIC (Pellom, 2001). Since the slides are given in advance, we manually add into the pronunciation dictionary the words that appear in slides but not in the pronunciation dictionary. To estimate similarity between a word vector (discussed earlier in Section 4.2) and an utterance, we sum up all keyword-spotting confidence scores assigned between them, normalize the resulted score by the length of the vector and the duration of the utterance, and then renormalize it to the range [0, 1] within the same spoken lecture.

We present the performance of our bullet-audio alignment model (AUDIO) in Figure 2 so that one can compare its effectiveness with the transcription based methods. The figure shows that the performance of the AUDIO model is comparable to the baseline transcription-based model, i.e., B-ALN, when the WERs of the transcripts are in the range of 37%–39%. The performance is comparable to the HG-ALN model when WERs are in the range of 42%–44%. Also, this suggests that incorporating hierarchical and global features compensates for the performance degradation of speech recognition in this range when the WER is 4%-6% higher.

Note that we did not observe that the performance is different when incorporating hierarchical information and global word distributions into the AUDIO model, so the AUDIO results in Figure 2 are the performance of both types of methods. The current keyword spotting component yields a high false-positive rate; e.g., it incorrectly reports many words that are acoustically similar to parts of other words that really appear in an utterance. This happened even when a high threshold is set. The noise impairs the benefit of hierarchical and distribution features.

## 7 Conclusions and discussions

This paper investigates the problem of imposing a known hierarchical structure onto an unstructured spoken document. Results show that incorporating local hierarchical constraints and global word distributions in the efficient dynamic programming framework yields a better performance over the baseline. Further experiments on a wide range of WERs confirm that the improvement is consistent, and show that both types of models are sensitive to speech recognition errors, particularly when WER increases to 30% and above. Moreover, directly imposing hierarchical structures onto raw speech through keyword spotting achieves competitive performance.

## References

Beeferman, D., A. Berger, and J. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

Branavan, S., Deshpande P., and Barzilay R. 2007. Generating a table-of-contents: A hierarchical discriminative approach. In *Proc. of Annual Meeting of the Association for Computational Linguistics*.

Chen, Y. and W. J. Heng. 2003. Automatic synchronization of speech transcript and slides in presentation. In *Proc. International Symposium on Circuits and Systems*.

Christensen, H., B. Kolluru, Y. Gotoh, and S. Renals. 2004. From text summarisation to style-specific summarisation for broadcast news. In *Proc. of the 26th European Conference on Information Retrieval*, pages 223–237.

Fan, Q., K. Barnard, A. Amir, A. Efrat, and M. Lin. 2006. Matching slides to presentation videos using sift and scene background. In *Proc. of ACM International Workshop on Multimedia Information Retrieval*, pages 239–248.

Garofolo, J., G. Auzanne, and E. Voorhees. 2000. The trec spoken document retrieval track: A success story. In *Proc. of Text Retrieval Conference*, pages 16–19.

Glass, J., T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay. 2007. Recent progress in the mit spoken lecture processing project. *Proc. of Annual Conference of the International Speech Communication Association*, pages 2553–2556.

Hearst, M. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Hori, C. and S. Furui. 2003. A new approach to automatic speech summarization. *IEEE Transactions on Multimedia*, 5(3):368–378.

Hsu, B. and J. Glass. 2006. Style and topic language model adaptation using hmm-lda. In *Proc. of Conference on Empirical Methods in Natural Language Processing*.

Jing, H. 2002. Using hidden markov modeling to decompose human-written summaries. *Computational Linguistics*, 28(4):527–543.

Leeuwis, E., M. Federico, and M. Cettolo. 2003. Language modeling and transcription of the ted corpus lectures. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*.

Liu, T., R. Hjelsvold, and J. R. Kender. 2002. Analysis and enhancement of videos of electronic slide presentations. In *Proc. IEEE International Conference on Multimedia and Expo*.

Malioutov, I., A. Park, B. Barzilay, and J. Glass. 2007. Making sense of sound: Unsupervised topic segmentation over acoustic input. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 504–511.

Marcu, D. 2000. The theory and practice of discourse parsing and summarization. The MIT Press.

Maskey, S. 2008. *Automatic Broadcast News Speech Summarization*. Ph.D. thesis, Columbia University.

Munteanu, C., R. Baecker, G. Penn, E. Toms, and E. James. 2006. Effect of speech recognition accuracy rates on the usefulness and usability of webcast archives. In *Proc. of ACM Conference on Human Factors in Computing Systems*, pages 493–502.

Munteanu, C., G. Penn, and R. Baecker. 2007. Web-based language modelling for automatic lecture transcription. In *Proc. of Annual Conference of the International Speech Communication Association*.

Murray, G. 2008. *Using Speech-Specific Characteristics for Automatic Speech Summarization*. Ph.D. thesis, University of Edinburgh.

Pellom, B. L. 2001. Sonic: The university of colorado continuous speech recognizer. *Tech. Rep. TR-CSLR-2001-01, University of Colorado*.

Pevsner, L. and M. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics,* 28:19–36.

Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. 2007. Numerical recipes: The art of science computing. Cambridge University Press.

Ruddarraju, R. 2006. *Indexing Presentations Using Multiple Media Streams*. Ph.D. thesis, Georgia Institute of Technology. M.S. Thesis.

Stark, L., S. Whittaker, and J. Hirschberg. 2000. Finding information in audio: A new paradigm for audio browsing and retrieval. In *Proc. of International Conference on Spoken Language Processing*.

Wang, F., C. W. Ngo, and T. C. Pong. 2003. Synchronization of lecture videos and electronic slides by video text analysis. In *Proc. of ACM International Conference on Multimedia*.

Zechner, K. and A. Waibel. 2000. Minimizing word error rate in textual summaries of spoken language. In *Proc. of Applied Natural Language Processing Conference and Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 186–193.

Zhu, X., X. He, C. Munteanu, and G. Penn. 2008. Using latent dirichlet allocation to incorporate domain knowledge for topic transition detection. In *Proc. of Annual Conference of the International Speech Communication Association*.

Zhu, X. 2010. *Summarizing Spoken Documents Through Utterance Selection*. Ph.D. thesis, University of Toronto.