Indexing Spoken Documents with Hierarchical Semantic Structures: Semantic Tree-to-string Alignment Models

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 addresses a semantic tree-to-string alignment problem: indexing spoken documents with known hierarchical semantic structures, with the goal to help index and access such archives. We propose and study a number of alignment models of different modeling capabilities and time complexities to provide a comprehensive understanding of these unsupervised models and hence the problem itself.

1 Introduction

The inherent difficulties in efficiently accessing spoken documents raise the need for ways to better organize such archives. Such a need parallels with the consistently increasing demand for and availability of audio content on web pages and other digital media, which, in turn, should come as no surprise, with speech being one of the most basic, most natural forms of human communication.

When intended to be read, written documents are almost always presented as more than uninterrupted text strings; e.g., indicative structures such as section/subsection headings and tables-of-contents are standard constituents created manually to help readers, whereas structures of this kind are rarely aligned with spoken documents, which has raised little concern—in most time of history, speech has not been ready for *navigation*, until very recently, when recording, delivering, and even automatic transcription were possible.

Navigating audio documents is often inherently much more difficult than browsing text. An obvious solution, in relying on human beings' ability of reading text, is to conduct a speech-to-text conversion through ASR, which in turn raises a new set of problems to be considered. First, the convenience and efficiency of reading transcripts are affected by errors produced in transcription channels, though if the goal is only to browse the most salient parts, recognition errors in excerpts can be reduced by considering ASR confidence (Xie and Liu, 2010; Hori and Furui, 2003; Zechner and Waibel, 2000) and the quality of excerpts can be improved from various perspectives (Zhang et al., 2010; Xie and Liu, 2010; Zhu et al., 2009; Murray, 2008; Zhu and Penn, 2006; Maskey and Hirschberg, 2005). Even if transcription quality were not a problem, browsing lengthy transcripts is not straightforward, since, as mentioned above, indicative browsing structures are barely manually created for and aligned with spoken documents. Ideally, such semantic structures should be inferred directly from the spoken documents themselves, but this is known to be difficult even for written texts, which are often more linguistically well-formed and less noisy than automatically transcribed text. This paper studies a less ambitious problem: we align an already-existing hierarchical browsing structure, e.g., the electronic slides of lecture recordings, with the sequential transcripts of the corresponding spoken documents, with the aim to help index and access such archives. Specifically, we study a number of semantic tree-to-string alignment models with different modeling capabilities and time complexities in order to obtain a comprehensive understanding of these models and hence the indexing task itself.

Semantic Structures of Spoken Documents Much previous work, similar to its written-text counterpart, has attempted to find certain *flat* structures of spoken documents such as topic and slide boundaries (Malioutov et al., 2007; Zhu et al., 2008), which, however, involve no hierarchical structures of a spoken document, thought as will be shown in this paper, topic-segmentation models can be considered in our alignment task. Research has also resorted to other multimedia channels, e.g., video (Fan et al., 2006), to detect slide

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 509–517,
Chiang Mai, Thailand, November 8 – 13, 2011.

transitions. This type of approaches, however, are unlikely to recover semantic structures more detailed than slide boundaries.

Zhu et al. (2010) investigate the problem of aligning electronic slides with lecture transcripts by first sequentializing bullet trees on slides with a pre-order walk before conducting alignment, through which the problem is reduced to a string-to-string alignment problem and conventional methods such as DTW (dynamic time warping) based alignment can then be directly applicable. A pre-order walk of bullet tree on slides is actually a natural choice, since speakers of presentations often follow such an order to develop their talks, i.e., they discuss a parent bullet first and then each of its children in sequence. However, although some remedies may be taken (Zhu et al., 2010), sequentializing the hierarchies before alignment, in principle, enforces a full linearity/monotonicity between transcripts and slide trees, which violates some basic properties of the problem that we will discuss. More recently, the work of (Zhu, 2011) proposes a graph-partitioning based model (revisited in Section 4) and shows that the model outperforms a bullet-sequentializing model.

With this previous work available, several important questions, however, are still open in obtaining a comprehensive understanding of the semantic tree-to-string alignment task. First of all, a basic question is associated with different ways of exploiting the semantic trees when performing alignment, which, as will be studied comprehensively in this paper, results in models of different modeling capabilities and time complexities. Second, all the models discussed above consider only similarities between bullets and transcribed utterances, while similarities among utterances, which directly underline a cohesion model, are generally ignored. We will show in this paper that the state-of-the-art topic-segmentation model (Malioutov and Barzilay, 2006) can be inherently incorporated into the graph-partitioning-based alignment models. Third, the different alignment objectives, e.g., that of the graph-partitioning models versus that of basic DTW-based models, are entangled together with different ways of exploiting the bullet tree structures in (Zhu, 2011). In this paper, we discuss two more quadratic-time models to bridge the gap.

Specifically, this paper studies nine different models, with the aim to provide a comprehensive understanding of the questions discussed above. In the remainder of the paper, we will first review the related work (Section 2) and more formally describe our problem (Section 3). Then we revisit the graph-partitioning alignment model (Section 4), before present all the alignment models we will study (Section 5). We describe our experiment setup in Section 7 and results in Section 8, and draw our conclusions in Section 9.

2 Related Work

Alignment of parallel texts In general, research on finding correspondences between parallel texts pervades both spoken and written language processing, e.g., in training statistical machine translation models, identifying relationship between human-written summaries and their original texts (Jing, 2002), force-aligning speech and transcripts in ASR, and grounding text with database facts (Snyder and Barzilay, 2007; Chen and Mooney, 2008; Liang et al., 2009). Our problem here, however, is distinguished in several major aspects, which need to be considered in our modeling. First, it involves segmentation—alignment is conducted together with the decision of the corresponding segment boundaries on transcripts; in other words, we are not finally concerned with the specific utterances that a bullet is aligned to, but the region of utterances. In such a sense, graph partitioning seems intuitively to be more relevant than models optimizing a full-alignment score. Second, unlike a string-to-string alignment task, the problem involves hierarchical tree structures. This allows for different ways of combining tree traversal with the alignment process, as will be studied in detail in this paper. Third, the hierarchical structures as well as the texts on them are fixed and unique to each document (here a lecture) and knowledge is little generalizable across different documents. We accordingly keep our solution in an unsupervised framework. Fourth, the length of transcripts and that of the hierarchies are very imbalanced, and the former can be as long as tens of thousands of utterances or hundreds of thousands of words, which requires a careful consideration of a model's time complexity.

Building Tables-of-contents on Written Text Learning semantic structures of written text has been studied in a number of specific tasks, which include, but not limited to, those finding semantic representations for individual sentences and

those constructing hierarchical structures among sentences or larger text blocks. A notable effort of the latter kind, for example, is the work of (Branavan et al., 2007), which aims at the ultimate goal of building tables-of-contents for written texts, though the problem was restricted to generating titles for each text span by assuming the availability of the structures of tables-of-contents and their alignments with text spans. Our work here can be thought of as an inverse problem, in which a specific type of semantic hierarchical structures are known, and we need to establish their correspondence with the spoken documents.

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 of (Marcu, 2000). However, when the task involves the understanding of high-level discourse, it becomes more challenging than 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, speech recognition errors, absence of formality and thematic boundaries, and less linguistically well-formedness of the spoken language, will further impair the conditions on which an reliable discourse-analysis algorithm is often built. In this paper, we study a less ambitious but naturally occurring problem.

3 Problem

We are given a speech sequence $U = u_1, u_2, ..., u_N$, where u_i is an utterance, and the corresponding hierarchical structure, which, in our work here, is a sequence of lecture slides containing a set of slide titles and bullets, $B = \{b_1, b_2, ..., b_M\}$, 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_m in the slides. Ψ is the edge set. With the definitions, our task is herein to find the triple (b_i, u_j, u_k), denoting that a bullet b_i is mapped to a region of lecture transcripts that starts from the jth utterance u_j and ends at the kth, inclusively. Constrained by the tree structure, the transcript region 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_n in the tree, the acquired boundary triples (b_i, u_{j_1}, u_{k_1}) and (b_n, u_{j_2}, u_{k_2}) should satisfy $j_1 \leq j_2$ and $k_1 \geq k_2$.

4 Graph-partitioning Models: A Revisit

To facilitate our discussion later in this paper, we briefly revisit the graph-partitioning alignment model proposed in (Zhu, 2011), which, inspired by (Malioutov and Barzilay, 2006; Shi and Malik, 2000), extended a graph-partitioning model to find the correspondence between the bullets on electronic slides and transcribed utterances.

Consider a general, simple two-set partitioning case, in which a boundary is placed on a graph $G = (V, E)$ to separate its vertices V into two sets, A and B, with all the edges between these two sets being removed. The objective, as we have mentioned above, is to minimize the following normalized-cut score:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \quad (1)$$

In equation (1), $cut(A, B)$ is the total weight of the edges being cut, i.e., those connecting A with B, while $assoc(A, V)$ and $assoc(B, V)$ are the total weights of the edges that connect A with all vertices V, and B with V, respectively. In general, minimizing such a normalized-cut score has been shown to be NP-complete. In our problem, however, the solution is constrained by the linearity of segmentation on transcripts, similar to that in topic segmentation (Malioutov and Barzilay, 2006). In such a situation, a polynomial-time algorithm exists (Zhu, 2011).

Consider a set of sibling bullets, $b_1, ..., b_m$, that appear on the same level of a bullet tree and share the same parent b_p. For the time being, we assume the corresponding region of transcripts has already been identified for b_p, say $u_1, ..., u_n$. We connect each bullet in $b_1, ..., b_m$ with utterances in $u_1, ..., u_n$ by their similarity, which results in a bipartite graph. Our task here is to place $m - 1$ boundaries onto the bipartite graph to partition the graph into m bipartite graphs and obtain triples, e.g., (b_i, u_j, u_k), to align b_i to $u_j, ..., u_k$, where $b_i \in \{b_1, ..., b_m\}$ and $u_j, u_k \in \{u_1, ..., u_n\}$ and $j <= k$. Since we have all descendant bullets to

help the partitioning, when constructing the bipartite graph, we actually include also all descendant bullets of each bullet b_i, but ignoring their orders within each b_i. We find optimal normalized cuts in a dynamic-programming process with the following recurrence relation:

$$C[i,k] = \min_{j \leq k} \{C[i-1,j] + D[i,j+1,k]\} \quad (2)$$

In equation (2), $C[i,k]$ is the optimal/minimal normalized-cut value of aligning the first i sibling bullets, $b_1, ..., b_i$, with the first k utterances, $u_1, ..., u_k$. It is computed by updating $C[i-1,j]$ with $D[i,j+1,k]$, for all possible j s.t. $j \leq k$, where $D[i,j+1,k]$ is a normalized-cut score for the triple (b_i, u_{j+1}, u_k) and is defined as follows:

$$D[i,j+1,k] = \frac{cut(A_{i,j+1,k}, V \setminus A_{i,j+1,k})}{assoc(A_{i,j+1,k}, V)} \quad (3)$$

where $A_{i,j+1,k}$ is the vertex set that contains the bullet b_i (including its descendant bullets, if any, as discussed above) and the utterances $u_{j+1}, ..., u_k$; $V \setminus A_{i,j+1,k}$ is its complement set.

Different from the topic segmentation problem (Malioutov and Barzilay, 2006), the graph-partitioning alignment model needs to remember the normalized-cut values between any region $u_j, ..., u_k$ and any bullet b_i in our task, which requires to use the additional subscript i in $A_{i,j+1,k}$, while in topic segmentation, the computation of both $cut(.)$ and $assoc(.)$ is only dependant on the left boundary j and right boundary k. Also, the similarity matrix here is not symmetric as in topic segmentation, but m by n, where m is the number of bullets, while n is the number of utterances.

As far as time complexity is concerned, the graph-partitioning models discussed above are quadratic with regards to N, i.e., $O(MN^2)$, where $M \ll N$; M and N denoting the number of bullets and utterances, respectively, with the loop kernel computing and filling $D[i,j,k]$ in equation 3, which is a $M \times N \times N$ matrix. Zhu (2011) applied the algorithm deterministically in traversing a bullet tree top-down: starting from the root, the normalized-cut algorithm finds the corresponding regions of transcripts for all the direct children of the root, fixes the regions, and repeats this process recursively to partition lower-level bullets. This whole algorithm is still quadratic $O(MN^2)$ but outperforms a bullet-sequentializing baseline.

5 Alignment Models

Now, we discuss the models that we will study further in this paper to address the problems rise earlier in the introduction section.

5.1 The $O(MN^4)$ Models

As discussed, Zhu (2011) proposed a graph-partitioning alignment model and applied it in a deterministic way along with a top-down traversal of bullet trees. Though such models could be very competitive in performance, an important question, however, is with regard to the performance of models that can optimize a global score rather than local ones on each set of sibling bullets, which requires a study of models with more modeling capability (containing the deterministic hierarchical models as a special case) and with higher time complexities.

Naively, searching all possible partitions to optimizing a global score needs to consider an exponential space in terms of the number of transcribed utterances, while applying dynamic programming similar to those used in syntactic parsing would keep the solution to be polynomial. In this section, we introduce such alignment models; or in another viewpoint, we formulate the alignment task in a parsing-like setting. A dynamic programming approach, e.g., that used in a conventional CYK parser, can be adapted to solve this problem, in which one can replace the splitter moving in each text span in the classic CYK with the quadratic bipartite-graph partitioning model discussed above. However, in our task here, the trees, unlike in a general parsing task, are given and fixed, meaning that the cells of a parsing table can be filled in a fixed order, i.e., a post order, so that the search speed can be improved by some constant.

Figure 1 shows an algorithm, in which we insert the bipartite graph partitioning model that works on sibling bullets (as discussed in Section 4) into a parsing search process (line (12)). We call this model **PrsCut**. Note that there are more than one way to conducting such a search, but they should yield the same results once the objective function, e.g., the normalized-cut score here, is the same.

Specifically, the *Main* function in Figure 1 takes as input an $M \times N$ similarity matrix, where, same as before, M and N denote the number of bullets and transcribed utterances in a lecture, respectively. The $Main$ function first computes the

```
Main
Input: simMat, an MxN similarity matrix
       root, the root of a bullet tree
Output: boundPos, bullets' boundaries on transcripts
1: cutCostTab = Cal_CutCostTab(simMat);
2: Build_Parsing_Tab(root, cutCostTab, prsTab);
3: boundPos = Decoding(root, prsTab);

Build_Parsing_Tab(curNd, cutCostTab, prsTab)
Input: curNd, current node/bullet in concern
       cutCostTab, an MxNxN normalized-cut cost table
Output: prsTab, an MxNxN parsing table
4: If current node curNd is a leaf then
5:    prsTab[curNd,:,:] = cutCostTab[curNd,:,:];
6: else
7:    for each child c^i of curNd do
8:       Build_Parsing_Tab(c^i, cutCostTab, prsTab);
9:    end for
10:   for i = 1...N do
11:      for j = i...N do
12:         bestScr = Bigraph_Alignment(curNd, prsTab, i, j);
13:         prsTab[curNd,i,j] = cutCostTab[curNd,i,j]+w*bestScr;
14:      end for
15:   end for
16: end if
```

Figure 1: An algorithm of optimizing a global normalized-cut score.

$cutCostTab$, which saves the $D[i, j+1, k]$ values defined by equation (3). Then the parsing table $prsTab$ is built with a post-order traversal algorithm *Build-Parsing-Tab*, followed by a decoding process that finds the optimal partitioning tree. As sketched in Figure 1, the *Build-Parsing-Tab* algorithm builds a 3-dimensional table $prsTab$, each cell saving a value that linearly combines the corresponding $cutCostTab$ value of the current node/bullet *curNd* and the optimal partitioning score $bestScr$ value calculated on its descendent, if any (see line (13)); or if the current node $curNd$ is a leaf itself, its $bestScr$ score is zero; in such case, the $prsTab$ value is initialized with the $cutCostTab$ value (line (5)). The recursive algorithm traverse the bullet tree in a post-order walk, which, as discussed above, utilizes the given, fixed bullet tree structures to fill the parsing table $prsTab$. The weight w in line (13) is set in a held-out data and note that if w is set to be 0, the model degrades to be the deterministic hierarchical model discussed in (Zhu, 2011) and referred to as $HieCut$ below in Section 5.2, since in this case the $prsTab$ is same as $costCustTab$. As far as time complexity is concerned, the whole algorithm is $O(MN^4)$, shown by the nested *for*-loops of line (10)-(15) that contain the $O(MN^2)$ bigraph-partitioning alignment in line (12). Similarly, we can insert a standard DTW-based alignment model into the line (12) here, which we call the **PrsBase** model. Note that the real algorithm is a little more complicated; e.g., we need to allow a parent bullet to have a different starting position than its first child, same as in (Zhu, 2011).

5.2 The $O(MN^2)$ models

Sequential Alignment Models As discussed earlier, in a simplified situation, our problem here can be formulated as a sequential alignment problem, based on a fairly reasonable assumption (Zhu et al., 2010): a speaker follows a pre-order walk of a bullet tree to develop the talk, i.e., discussing a parent bullet first, followed by each of its children in sequence. Accordingly, the models first sequentialize bullet trees with a pre-order walk before conducting alignment, through which the problem is reduced to a string-to-string alignment problem and conventional methods such as DTW-like alignment can then be applicable. Such a pre-order walk has also been assumed by (Branavan et al., 2007) to reduce the search space in their table-of-contents generation task, a problem in which a tree hierarchy has already been aligned with a span of written text, while the title of each node on the tree needs to be generated.

With this formulation, we first included here the baseline model in (Zhu et al., 2010), which applies a typical DTW-based alignment. We refer to the model as **SeqBase**. In addition, we applied the graph-partitioning based models discussed in (Zhu, 2011) to align the sequentialized bullets and the corresponding transcribed utterances, and we call this model **SeqCut**. The motivation of studying $SeqCut$ is to further understand the benefit of graph-partitioning based models. For example, it allows us to disentangle the benefit of the deterministic graph-partitioning models in (Zhu, 2011): whether the benefit is due to the modeling advantage of the proposed partitioning objective or its avoiding sequentializing bullet trees.

In principle, sequentializing bullet trees before alignment enforces a full linearity/monotonicity between transcripts and these bullet trees, which, though based on a reasonable assumption and is fairly effective (as will be shown in our comprehensive comparison later), misses some basic properties of the problem. For example, the generative process of lecture speech, with regards to a hierarchical structure (here, bullet trees), is char-

acterized in general by a speaker's producing detailed content for each bullet when discussing it, during which sub-bullets, if any, are talked about recursively. By the nature of the problem, words in a bullet could be repeated multiple times, even when the speaker traverses to talk about the descendant bullets in the depth of the sub-trees. That is, the content of a bullet could be mentioned not only before its children but also very likely when the speaker traverses to talk descendant bullets, if any, which violate the pre-order-walk assumption.

Though with shortcomings, an important benefit of formulating the task as a sequential-alignment problem is its computational efficiency: solutions can be acquired in quadratic time. This is of particular importance for this task, considering that the length of a document, such as a lecture or a book, is often long enough to make less efficient algorithms practically intractable. A natural question to be ask is therefore whether we can, in principle, model the problem better, but still keep the time complexity quadratic, i.e., $O(MN^2)$.

Deterministic Hierarchical Models Deterministically deciding bullets' boundaries on transcribed utterances when traversing the bullet tree can keep the solution within a quadratic time complexity and avoid a sequentialization of bullet trees beforehand. For example, in (Zhu, 2011), the graph-partitioning alignment model, as discussed above, is applied in such a deterministic way; the model recursively traverses a bullet tree by first determining transcript boundaries of the direct children of the root, fixing the boundaries found, and then determining boundaries for the descendant bullets recursively[1]. We refer to this model as $HieCut$ in this paper. Note that though working deterministically, this models utilize the similarities associated with all descendant bullets of the current sibling bullets under concern, to find the optimal boundaries between these siblings. In addition, we include a standard DTW-based alignment model in such a deterministic-decision process, called the **HieBase** model in the remainder of this paper.

One major benefit of the deterministic hierarchical alignment models is their time complexity: still quadratic, same as the sequential alignment model discussed above, though models like $HieCut$ can achieve a very competitive performance, which we will discuss in detail later. Also, the deterministic hierarchical models need less memories than the corresponding $O(MN^4)$ models and even the sequential models. For example, the memory needed by $HieCut$ is proportional to the maximal number of sibling bullets in a tree, not the total number of bullets.

[1] A pre-order walk can be used here (not for sequentializing bullet trees though); other top-down transversing methods are also applicable, e.g., a breadth-first search, once a parent bullet is visited before its children.

6 The Topic-segmentation Model

Up to now, we have discussed a variety of alignment models with different model capabilities and time complexities, which, however, consider only similarities between bullets and utterances. Cohesion in text or speech, by itself, often evidenced by the change of lexical distribution (Hearst, 1997), can also indicate topic or subtopic transitions, even among subtle subtopics (Malioutov and Barzilay, 2006). In our problem here, when a lecturer discusses a bullet, the words used are likely to be different from those used in another bullet, suggesting that the spoken documents themselves, when ignoring the alignment model above for the time being, could potentially indicate the semantic boundaries that we are interested in here. Particularly, the cohesion conveyed by the repetition of the words that appear in transcripts but not in slides could be additionally helpful; this is very likely to happen considering the significant imbalance of text lengths between bullets and transcripts, from which the alignment models by themselves may suffer.

$$C[i,k] = \min_{j \le k} \{C[i-1,j] + \lambda_1 D[i,j+1,k] + (1-\lambda_1) S[j+1,k]\} \quad (4)$$

where,

$$S[j+1,k] = \frac{cut(A_{j+1,k}, V \setminus A_{j+1,k})}{assoc(A_{j+1,k}, V)} \quad (5)$$

In fact, a state-of-the-art topic-segmentation model (Malioutov and Barzilay, 2006) (also called a cohesion model in this paper) can be naturally incorporated into the graph-partitioning alignment models that we have discussed. That is, we can augment the $SeqCut$, $HieCut$, and $PrsCut$ models with the cohesion models to form three new models **SeqCutTpc**, **HieCutTpc**, and **PrsCutTpc**, respectively. To achieve this, we modify equation (2) to equation (4), where $S[j+1,k]$ is calculated as in (Malioutov and Barzilay,

2006), which denotes the normalized partition cost of the segment from utterance u_{j+1} to u_k, inclusively. For complexity, since the cohesion model is $O(MN^2)$, linearly combining it would not increase the time complexities of the corresponding polynomial alignment models, which are at least $O(MN^2)$ by themselves.

7 Experiment Set-up

Corpus Our experiment uses a corpus of four 50-minute university lectures taught by the same instructor, which contain 119 slides composed of 921 bullets. The automatic transcripts of the speech contain approximately 30,000 word tokens, roughly equal to 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, while students' comments and questions were not recorded. The speech is split into utterances by pauses longer than 200ms, resulting in around 4000 utterances. The slides and automatic transcripts of one lecture were used as the development set. In practice, each lecture is divided into three roughly equally-long pieces in all our experiments discussed below, for pragmatic computational consideration of calculating the $O(MN^4)$ models quickly enough.

Building the Graphs The transcripts were generated with the SONIC toolkit (Pellom, 2001), with the models trained as suggested by (Munteanu et al., 2007), in which one language model was trained on SWITCHBOARD and the other used also corpus obtained from the Web through searching the words on slides. Both bullets and automatic transcripts were stemmed with the Porter stemmer and stopwords were removed. The similarities between bullets and utterances and those between utterances were calculated with different distance metrics, i.e., cosine, exponential cosine (Malioutov and Barzilay, 2006) for topic segmentation, and a normalized word-overlapping score used in summarization (Radev et al., 2004), from which we chose the one (regular cosine) that optimizes our baseline. Our graph-partitioning models then used exactly the same setting. The lexical weighting is same as in (Malioutov et al., 2007), for which we split each lecture into M chunks, the number of bullets. Finally, we obtained a M-by-N bullet-utterance similarity matrix and a N-by-N utterance-utterance matrix to optimize the alignment model and topic-segmentation model, respectively, while M and N, as already mentioned, denote the number of bullets and utterances of a lecture, respectively.

Evaluation Metric The metric used in our evaluation is straightforward—automatically acquired boundaries on transcripts for each slide bullet are compared against the corresponding gold-standard boundaries to calculate offsets measured in number of words, counted after stopwords having been removed, which are then averaged over all boundaries to evaluate model performance. 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. Note that topic segmentation research often uses metrics such as P_k and *WindowDiff* (Malioutov and Barzilay, 2006; Beeferman et al., 1999; Pevsner and Hearst, 2002). Our problem here, as an alignment problem, has an exact 1-to-1 correspondence between a gold and automatic boundary, in which we can directly measure the exact offset of each boundary.

8 Experimental Results

Alignment Models Table 1 presents the experimental results obtained on the automatic transcripts generated by the ASR models discussed above, with WERs of 0.43 and 0.48, respectively, which are typical for lectures and conference presentations in realistic and less controlled situations (Leeuwis et al., 2003; Hsu and Glass, 2006; Munteanu et al., 2007).

The results show that among the four quadratic models, i.e., the first four models in the table, $HieCut$ achieves the best performance. The results also suggest that the improvement of $HieCut$ over $SeqBase$ comes from two aspects. First, the normalized-cut objective used in the graph-partitioning based model seems to outperform that used in the baseline, indicated by the better performance of $SeqCut$ over $SeqBase$, since both take as input the same, sequentialized bullet sequence and the corresponding transcribed utterances. The DTW-based objective used in $SeqBase$ corresponds to finding the optimal path that maximizes the similarity score between the bullet sequence and the transcripts. Second, the better performance of $HieCut$ and $SeqCut$ shows that $HieCut$ further benefits from avoiding sequentializing the bullet trees. However, this two

aspects of benefit do not come independently, since the former (performance of an alignment objective) can significantly affect the latter (whether a model can benefit from avoiding sequentializing bullets). This is evident in the inferior performance of $HieBase$. Manual analysis of its errors shows that $HieBase$ is less accurate than $HieCut$ on higher-level bullets and the errors in turn severely impair the decisions made on lower-level bullets in the deterministic decision process: the errors propagate severely in such a deterministic process.

Models	WER=0.43	WER=0.48
SeqBase	15.19	18.44
SeqCut	12.87	16.16
HieBase	21.06	24.25
HieCut	12.13	15.95
PrsBase	15.05	18.18
PrsCut	12.05	15.20

Table 1: The performances of different alignment models.

A closer examination of errors made by $HieBase$ suggests that in a DTW-based alignment, a large subtree is likely to be aligned to a region larger than it should be, particularly for higher-level bullets (e.g., slides), where the subtree sizes vary more, e.g., some slides containing much textual content and others containing little. It seems that $HieCut$ could counteract this effect with its capability of normalizing partition sizes (see the denominators in both equation (1) and (3)). The usefulness of the normalization has also been discussed in other tasks such as image segmentation (Shi and Malik, 2000). Compared with those of $HieBase$, segments in the $SeqBase$ model are smaller (all non-leaf bullets do not include its descendants after being sequentialized) and the pre-order walk constrains the alignment range of bullets, which often avoid errors of long offsets. Again, the $HieCut$ model is quadratic in time, it uses less memories than the $O(MN^4)$ models and even the $SeqCut$ model, and it achieves a very competitive overall performance.

The results in Table 1 also shows that the ($O(MN^4)$) models, which conduct a more thorough search, improve the performance in all situations.

Effect of Topic-segmentation Models The effect of the topic-segmentation model is presented in Table 2. To facilitate reading, we also copy here the relevant results from Table 1. The results show that incorporating text cohesion additionally reduces the errors consistently for all models, though the specific improvement varies.

Models	WER=0.43	WER=0.48
SeqCut	12.87	16.16
SeqCutTpc	12.77	15.14
HieCut	12.13	15.95
HieCutTpc	11.82	15.28
PrsCut	12.05	15.20
PrsCutTpc	11.34	14.62

Table 2: The effect of topic-segmentation models.

9 Conclusions

In addressing the semantic tree-to-string alignment problem described, this paper proposes and studies a number of models with different modeling capabilities and time complexities. Experimental results show that among the quadratic alignment models ($O(MN^2)$), $HieCut$ consistently achieves the best performance, while the $O(MN^4)$ models that optimize a global objective score further improve the performance, though such models are, pragmatically, much more computationally expensive. This paper also relates alignment models with topic-segmentation models by showing that a state-of-the-art topic-segmentation models can be inherently incorporated into the graph-partitioning based alignment models. The experimental results show the benefit of considering such cohesion knowledge.

References

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

S. Branavan, 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*.

D.L. Chen and R.J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proc. of International Conference on Machine Learning*.

Q. Fan, 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.

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

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

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

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

E. Leeuwis, 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*.

P. Liang, M. Jordan, and D. Klein. 2009. Learning semantic correspondences with less supervision. In *Proc. of Annual Meeting of the Association for Computational Linguistics*.

I. Malioutov and R. Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proc. of International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics*.

I. Malioutov, A. Park, R. 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.

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

S. Maskey and J. Hirschberg. 2005. Comparing lexial, acoustic/prosodic, discourse and structural features for speech summarization. In *Proc. of European Conference on Speech Communication and Technology*, pages 621–624.

C. Munteanu, 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*.

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

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

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

D. Radev, H. Jing, M. Stys, and D. Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40:919–938.

J. Shi and J. Malik. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22.

B. Snyder and R. Barzilay. 2007. Database-text alignment via structured multilabel classification. In *Proc. of International Joint Conference on Artificial Intelligence*.

S. Xie and Y. Liu. 2010. Using confusion networks for speech summarization. In *Proc. of International Conference on Human Language Technology and Annual Meeting of North American Chapter of the Association for Computational Linguistics*.

K. Zechner 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.

J. Zhang, H. Chan, and P. Fung. 2010. Extractive speech summarization using shallow rhetorical structure modeling. *IEEE Transactions on Audio, Speech and Language Processing*, 18:1147–1157.

X. Zhu and G. Penn. 2006. Summarization of spontaneous conversations. In *Proc. of International Conference on Spoken Language Processing*, pages 1531–1534.

X. Zhu, 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*.

X. Zhu, G. Penn, and F. Rudzicz. 2009. Summarizing multiple spoken documents: Finding evidence from untranscribed audio. In *Proc. of Annual Meeting of the Association for Computational Linguistics*.

X. Zhu, C. Cherry, and G. Penn. 2010. Imposing hierarchical browsing structures onto spoken documents. In *Proc. of International Conference on Computational Linguistics*.

X. Zhu. 2011. A normalized-cut model for aligning hierarchical browsing structures with spoken documents. In *Proc. of the Fifteenth Conference on Computational Natural Language Learning (to appear)*.