A normalized-cut alignment model for mapping hierarchical semantic structures onto spoken documents

Zhu, Xiaodan

Publisher's version / Version de l'éditeur:

Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL '11), pp. 210-218, 2011-07-01

A Normalized-Cut Alignment Model for Mapping Hierarchical Semantic Structures onto Spoken Documents

Xiaodan Zhu
Institute for Information Technology
National Research Council Canada
Xiaodan.Zhu@nrc-cnrc.gc.ca

Abstract

We propose a normalized-cut model for the problem of aligning a known hierarchical browsing structure, e.g., electronic slides of lecture recordings, with the sequential transcripts of the corresponding spoken documents, with the aim to help index and access the latter. This model optimizes a normalized-cut graph-partitioning criterion and considers local tree constraints at the same time. The experimental results show the advantage of this model over Viterbi-like, sequential alignment, under typical speech recognition errors.

1 Introduction

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 (Ge and Mooney, 2005; Zettlemoyer and Collins, 2005; Lu et al., 2008), and those constructing hierarchical structures among sentences or larger text blocks (Marcu, 2000; Branavan et al., 2007). The inverse problem of the latter kind, e.g., aligning certain form of already-existing semantic hierarchies with the corresponding text sequence, is not so much a prominent problem for written text as it is for spoken documents. In this paper, we study a specific type of such a problem, in which a hierarchical browsing structure, i.e., electronic slides of oral presentations, have already existed, the goal being to impose such a structure onto the transcripts of the corresponding speech, with the aim to help index and access spoken documents as such.

Navigating audio documents is often inherently much more difficult than browsing text; an obvious solution, in relying on human beings' ability to read text, is to conduct a speech-to-text conversion through automatic speech recognition (ASR). Implicitly, solutions as such change the conventional speaking-for-hearing construals: now speech can be *read* through its transcripts, though, in most cases, it was not intended for this purpose, which in turn raises a new set of problems.

The convenience and efficiency of reading transcripts (Stark et al., 2000; Munteanu et al., 2006) are first affected by errors produced in transcription channels for various reasons, though if the goal is only to browse salient excerpts, recognition errors on the extracts can be reduced by considering ASR confidence scores (Xie and Liu, 2010; Hori and Furui, 2003; Zechner and Waibel, 2000): trading off the expected salience of excerpts with their recognition-error rate could actually result in the improvement of excerpt quality in terms of the amount of important content being correctly presented (Zechner and Waibel, 2000).

Even if transcription quality were 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, even when existing, are rarely aligned with spoken documents completely.

This paper studies the problem of imposing a

known hierarchical browsing structure, e.g., the electronic slides of lecture recordings, onto the sequential transcripts of the corresponding spoken document, with the aim to help index and hence access the latter more effectively. Specifically, we propose a graph-partitioning approach that optimizes a normalized-cut criterion globally, in traversing the given hierarchical semantic structures. The experimental results show the advantage of this model over Viterbi-like, sequential alignment, under typical speech recognition errors.

2 Related work

Flat 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. 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 of a spoken document. Research has also resorted to other multimedia channels, e.g., video (Liu et al., 2002; Wang et al., 2003; Fan et al., 2006), to detect slide transitions. This type of research, however, is unlikely to recover semantic structures in more details than slide boundaries.

Hierarchical structures of spoken documents Recently, research has started to align hierarchical browsing structures with spoken documents, given that inferring such structures directly from spoken documents is still too challenging. Zhu et al. (2010) investigates bullet-slide alignment by first sequentializing bullet trees with a pre-order walk before conducting alignment, through which the problem is reduced to a string-to-string alignment problem and an efficient Viterbi-like method can be naturally applied. In this paper, we use such a sequential alignment as our baseline, which takes a standard dynamic-programming process to find the optimal path on an M-by-N similarity matrix, where M and N denote the number of bullets and utterances in a lecture, respectively. Specifically, we chose the path that maps each bullet to an utterance to achieve the highest total bullet-utterance similarity score; this path can be found within a standard $O(MN^2)$ time complexity.

A pre-order walk of the hierarchical tree is a natural choice, since speakers of presentations often follow such a order in developing their talk; i.e., they often talk about a bullet first and then each of its children in sequence. A pre-order walk is also assumed by Branavan et al. (2007) in their table-of-content generation task, a problem in which a hierarchical structure has already been assumed (aligned) with a span of written text, but the title of each node needs to be generated.

In principle, such a sequential-alignment approach allows a bullet to be only aligned to one utterance in the end, which does not model the basic properties of the problem well, where the content in a bullet is often repeated not only when the speaker talks about it but also, very likely, when he discusses the descendant bullets. Second, we suspect that speech recognition errors, when happening on the critical anchoring words that bridging the alignment, would make a sequential-alignment algorithm much less robust, compared with methods based on many-to-many alignment. This is very likely to happen, considering that domain-specific words are likely to be the critical words in deciding the alignment, but they are also very likely to be mis-recognized by an ASR system at the same time, e.g., due to out-of-vocabulary issue or language-model sparseness. We will further discuss this in more details later in our result section. Third, the hierarchical structures are lost in the sequentialization of bullets, though some remedy could be applied, e.g., by propagating a parent bullet's information onto its children (Zhu et al., 2010).

On the other hand, we should also note that the benefit of formulating the problem as a sequential alignment problem is its computational efficiency: the solution can be calculated with conventional Viterbi-like algorithms. This property is also important for the task, since the length of a spoken document, such as a lecture, is often long enough to make *inefficient* algorithms practically intractable.

An important question is therefore how to, in principle, model the problem better. The second is how time efficient the model is. Malioutov and Barzilay (2006) describe a dynamic-programming version

of a normalized-cut-based model in solving a topic segmentation problem for spoken documents. Inspired by their work, we will propose a model based on graph partitioning in finding the correspondence between bullets and the regions of transcripts that discuss them; the proposed model runs in polynomial time. We will empirically show its benefit on both improving the alignment performance over a sequential alignment and its robustness to speech recognition errors.

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, if not otherwise noted. $\aleph$ 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_i, u_{j_2}, u_{k_2}) should satisfy $j_1 \leq j_2$ and $k_1 \geq k_2$. Figure 1 shows a slide, its structure, and the correspondence between one of its bullets and a region of transcribed utterances (the root that concatenates all such slides of a lecture together is not shown here).

4 A graph-partitioning approach

The generative process of lecture speech, with regard to a hierarchical structure (here, bullet trees), is characterized 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 its 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. In principle, a model would be desirable to consider such properties between a slide bullet, including all its descendants, and utterance transcripts, as well as the constraints of bullet trees. We formulate the problem of finding the correspondence between bullets and transcripts as a graph-partitioning problem, as detailed below.

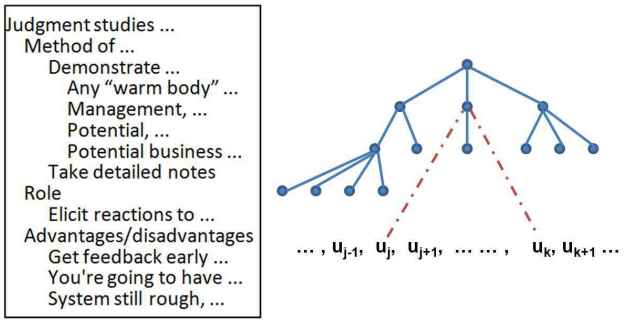

Figure 1: A slide, its tree structure, and the correspondence between one of its bullets and a region of transcribed utterances $(u_j, u_{j+1}..., u_k)$.

The correspondence between bullets and transcribed utterances is evidenced by the similarities between them. In a graph that contains a set of bullets and utterances as its vertices and similarities between them as its edges, our aim is to place boundaries to partition the graph into smaller ones in order to obtain triples, e.g., (b_i, u_j, u_k), that optimize certain criterion. Inspired by the work of (Malioutov and Barzilay, 2006; Shi and Malik, 2000), we optimize a normalized-cut score, in which the total weight of edges being cut by the boundaries is minimized, normalized by the similarity between the bullet b_i and the entire vertices, as well as between the transcript region $u_j, ..., u_k$ and the entire vertices, respectively.

Consider a simple two-set case first, 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:

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

where,

$$cut(A,B) = \sum_{a\in A, b\in B} w(a,b)$$
$$assoc(A,V) = \sum_{a\in A, v\in V} w(a,v)$$
$$assoc(B,V) = \sum_{b\in B, v\in V} w(b,v)$$

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; $w(a,b)$ is an edge weight between a vertex a and b.

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 (Malioutov and Barzilay, 2006). In such a situation, a polynomial-time algorithm exists. Malioutov and Barzilay (2006) describe a dynamic-programming algorithm to conduct topic segmentation for spoken documents. We modify the method to solve our alignment problem here, which, however, needs to cope with the bipartite graphs between bullets and transcribed sentences rather than symmetric similarity matrices among utterances themselves. We also need to integrate this in considering the hierarchical structures of bullet trees.

We first 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 $u_j, u_k \in \{u_1, ..., b_n\}$ 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 will revisit this in more details later. We find optimal normalized cuts in a dynamic-programming process with the following recurrence relation:

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

$$B[i,k] = \arg\min_{j\le k}\{C[i-1,j] + D[i,j+1,k]\} \quad (3)$$

In equation (2) and (3), $C[i,k]$ is the optimal/minimal normalized-cut value of aligning the first i sibling bullets, $b_1, ..., b_i$, with the first k utterances, $u_1, ..., b_k$, while $B[i,k]$ records the backtracking indices corresponding to the optimal path yielding the current $C[i,k]$. As shown in equation (2), $C[i,k]$ is computed by updating $C[i-1,j]$ with $D[i,j+1,k]$, for all possible j s.t. $j \le k$, 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:

$$D[i,j+1,k] = \frac{cut(A_{i,j+1,k}, V \setminus A_{i,j+1,k})}{assoc(A_{i,j+1,k}, V)} \quad (4)$$

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 et al., 2007), we need to remember the normalized-cut values between any region $u_j, ..., u_k$ and any bullet b_i in our task, so we need 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. Note that the similarity matrix here is not symmetric as it is in topic segmentation, but m by n, where m is the number of bullets, while n is the number of utterances.

For any triple (b_i, u_{j+1}, u_k), there are two different types of edges being cut: those between $B_{in} \stackrel{\text{def}}{=} \{b_i\}$ (again, including b_i and all its descendant bullets) and $U_{out} \stackrel{\text{def}}{=} \{u_1, ..., u_j, u_{k+1}, ..., u_m\}$, as well as those between $B_{out} \stackrel{\text{def}}{=} \{b_1, ..., b_{i-1}, b_{i+1}, ..., b_m\}$ and $U_{in} \stackrel{\text{def}}{=} \{u_{j+1}, ..., u_k\}$. We discriminate these two types of edges. Accordingly, $cut(.)$ and

$assoc(.)$ in equation (4) are calculated with equation (5) and (6) below by linearly combining the weights of these two types of edges with λ, whose value is decided with a small held-out data.

$$cut(A_{i,j+1,k}, V \setminus A_{i,j+1,k}) = \lambda \sum_{b \in B_{in}, u \in U_{out}} w(b,u) + (1-\lambda) \sum_{b' \in B_{out}, u' \in U_{in}} w(b',u') \quad (5)$$

$$assoc(A_{i,j+1,k}, V) = \lambda \sum_{b \in B_{in}, u \in V} w(b,u) + (1-\lambda) \sum_{b' \in U_{in}, u' \in V} w(b',u') \quad (6)$$

In addition, different form that in topic segmentation, where a segment must not be empty, we shall allow a bullet b_i to be aligned to an empty region, to model the situation that a bullet is not discussed by the speaker. To do so, we made j in equation (2) and (3) above to be able to equal to k in the subscript, i.e., $j \leq k$. Specifically, when $j = k$, the set $A_{i,j+1,k}$ has no internal edges, and $D[i, j+1, k]$ is either equal to 1, or often not defined if $assoc(A_{i,j+1,k}, V) = 0$. For the latter, we reset $D[i, j+1, k]$ to be 1.

A visual example of partitioning sibling bullets b_1, b_2, and b_3 is shown in Figure 2, in which the descendant bullets of them (here, b_4, b_5, and b_6) are also considered. Note that we only show direct children of b_1 here, while, as discussed above, all descendant bullets, if any, will be considered.

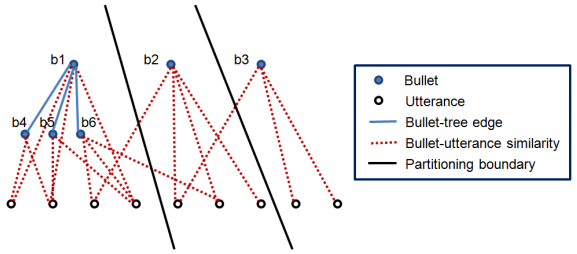

Figure 2: A visual example of partitioning sibling bullets b1, b2, and b3.

Up to now, we have only considered partitioning sibling bullets by assuming the boundaries of their parent on lecture transcripts have already been given, where the sibling bullets and the corresponding transcripts form a bipartite graph. When partitioning the entire bullet trees and all utterances for a lecture, the graph contains not only a bipartite graph but also the hierarchical trees themselves. We decouple this two parts of graph by a top-down traversal of the bullet trees: starting from the root, for each node on the bullet tree, we apply the normalized-cut algorithm discussed above to find the corresponding regions of transcripts for all its direct children, and repeat this process recursively. In each visit to partition a group of sibling bullets, to allow the first child to have a different starting point from its parent bullet (the speaker may spend some time on the parent bullet itself before talking about each child bullet), we inserted an extra child in front of the first child and copy the text of the parent bullet to it. Note that in each visit to partition a group of sibling bullets, the solution found is optimal on that level, which, again, results in a powerful model since all descendant bullets, if any, are all considered. For example, processing high-level bullets first is expected to benefit from the richer information of using all their descendants in helping find the boundaries on transcripts accurately. Recall that we have discussed above how to incorporate the descendant bullets into this process. It would also dramatically reduce the searching space of partitioning lower-level bullets.

As far as computational complexity is concerned, the graph-partitioning method discussed above is polynomial, $O(MN^2)$, with M and N denoting the number of bullets and utterances in a lecture, respectively. Note that M is often much smaller than N, $M \ll N$. In more details, the loop kernel of the algorithm is computing $D[i, j, k]$. This in total needs to compute $\frac{1}{2}(MN^2)$ values, which can be pre-calculated and stored before dynamic-programming decoding runs; the later, as normal, is $O(MN^2)$, too.

5 Experiment set-up

5.1 Corpus

Our experiment uses a corpus of four 50-minute third-year university lectures taught by the same instructor on the topics of human-computer interaction (HCI), which contain 119 slides composed of

921 bullets prepared by the lecturer himself. 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 held out to decide the value of λ in differentiating the two different types of edges being cut, as discussed in Section 4. The boundaries between adjacent slides were marked manually during the lectures were recorded, by the person who oversaw the recording process, while the boundaries between bullets within a slide were annotated afterwards by another human annotator.

5.2 Building the graphs

The lecture speech was first transcribed into text automatically with ASR models. The first ASR model is a baseline with its acoustic model trained on the WSJ0 and WSJ1 subsets of the 1992 development set of the Wall Street Journal (WSJ) dictation corpus, which contains 30 hours of data spoken by 283 speakers. The language model was trained on the Switchboard corpus, which contains 2500 telephone conversations involving about 500 English-native speakers, which was suggested to be suitable for the conversational style of lectures, e.g., by (Munteanu et al., 2007; Park et al., 2005). The whole model yielded a word error rate (WER) at 0.48. In the remainder of this paper, we call the model as ASR Model 1.

The second model is an advanced one using the same acoustic model. However, its language model was trained on domain-related documents obtained from the Web through searching the words appearing on slides, as suggested by Munteanu et al. (2007). This yielded a WER of 0.43, which is a typical WER for lectures and conference presentations (Leeuwis et al., 2003; Hsu and Glass, 2006; Munteanu et al., 2007), though a lower WER is possible in a more ideal condition (Glass et al., 2007), e.g., when the same course from the previous semester by the same instructor is available. The 3-gram language models were trained using the CMU-CAM Language Modelling Toolkit (Clarkson and Rosenfeld, 1997), and the transcripts were generated with the SONIC toolkit (Pellom, 2001). The out-of-vocabulary rates are 0.3% in the output of ASR Model 1 and 0.1% in that of Model 2, respectively.

Both bullets and automatic transcripts were stemmed and stop words in them were removed. We then calculated the similarity between a bullet and an utterance with the number of overlapping words shared, normalized by their lengths. Note that using several other typical metrics, e.g., cosine, resulted in a similar trend of performance change—our conclusions below are consistent under these situations, though the specific performance scores (i.e., word offsets) are different. Finally, the similarities between bullets and utterances yielded a single M-by-N similarity matrix for each lecture to be aligned, with M and N denoting the number of bullets in slides and utterances in transcripts, respectively.

5.3 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. The offset scores are 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 et al., 2007; 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.

6 Experimental results

Table 1 presents the experimental results obtained on the automatic transcripts generated by the ASR models discussed above, with WERs at 0.43 and 0.48, respectively, which are typical WERs for lectures and conference presentations in realistic and

less controlled situations. SEQ-ALN in the table stands for the Viterbi-like, sequential alignment discussed above in section 2, while G-CUT is the graph-partitioning approach proposed in this paper. The values in the table are the average word-offset scores counted after stop-words having been removed.

	WER=0.43	WER=0.48
SEQ-ALN	15.22	20.38
G-CUT	13.41	16.77
Offs. Reduction	12%	18%

Table 1: The average word offsets of automatic boundaries from the gold-standard.

Table 1 shows that comparing these two polynomial-time models, G-CUT reduces the average offsets of SEG-ALN under both WERs. On the transcripts with 0.48 WER, the average word-offset score is reduced by approximately 18% from 20.38 to 16.77, while for the transcripts with WER at 0.43, the offset reduction is 12%, from 15.22 to 13.41. Since both models use exactly the same input similarity matrices, the differences between their results confirm the advantage of the modeling principle behind the proposed approach. Although the graph-partitioning model could be extended further, e.g., with the approach in (Zhu et al., 2010), our primary interest here is the principle modeling advantage of this normalized-cut framework.

The results in Table 1 also suggest that the graph-partitioning model is more robust to speech recognition errors: when WERs increase from 0.43 to 0.48, the error of G-CUT increases by 25%, from 13.41 to 16.77, while that of SEQ-ALN increases by 44%, from 15.22 to 20.38. We due this to the fact that the graph-partitioning model considers multiple alignments between bullets, including their descendants, and the transcribed utterances, where mismatching between bullet and transcript words, e.g., that caused by recognition errors, is less likely to impact the graph-partitioning method, which bases its optimization criterion on multiple alignments, e.g., when calculating $cut(.)$ and $assoc(.)$ in equation (5) and (6). Recall that the ASR Model 2 includes domain-specific Web data to train the language models, which were acquired by using bullet words to search the Web. It is expected to increase the recognition accuracy on domain words, particularly those appearing on the slides. Therefore, Model 2 is likely to particularly increase the correct matching between bullets and transcript.

The results in Table 1 also show the usefulness of better ASR modeling on the structure-imposing task here. As discussed in the introduction section earlier, browsing automatic transcripts of long spoken documents, such as lectures, is affected by both speech recognition errors and lack of browsing structures. Table 1 shows that the improvement in solving the first problem also helps the second.

Last, from a pragmatic viewpoint of system development, the graph-partitioning algorithm is simple to implement: the essence of equation (2)-(6) is to find the optimal normalized-cut score characterized by computing $D[i, j + 1, k]$ and updating the formulae with it, which is not much more complicate to build than the baseline. Also, the practical speed difference between these two types of models is not obvious on our dataset.

7 Conclusion

This paper proposes a graph-partitioning approach for aligning a known hierarchical structure with the transcripts of the corresponding spoken document through optimizing a normalized-cut criterion. This approach models the basic properties of the problem and is quadratic-time. Experimental results show both its advantage on improving the alignment performance over a standard sequential-alignment baseline and its robustness to speech recognition errors, while both take as input exactly the same similarity matrices. From a pragmatic viewpoint of system development, this graph-partitioning-based algorithm is simple to implement. We believe immediate further work such as combining the normalized-cut model with CYK-like dynamic programing to traverse the semantic trees in alignment could help us further understand the problem, though such models need much more memory in practice if not properly optimized and have a higher time complexity. Also, topic-segmentation (cohesion) models can be naturally combined with the alignment model discussed here. We will study such problems as our immediate future work.

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*.

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

P. Clarkson and R. Rosenfeld. 1997. Statistical language modeling using the cmu-cambridge toolkit. In *Proc. of ISCA European Conf. on Speech Communication and Technology*, pages 2707–2710.

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.

R. Ge and R. J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proc. of Computational Natural Language Learnine*, pages 9–16.

J. Glass, 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.

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*.

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*.

T. Liu, 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*.

W. Lu, H. T. Ng, W. S. Lee, and L. S. Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *Proc. of Empirical Methods in Natural Language Processing*, pages 783–792.

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.

C. Munteanu, 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.

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*.

A. Park, T. Hazen, and J. Glass. 2005. Automatic processing of audio lectures for information retrieval. In *Proc. of IEEE Conf. on Acoustics, Speech, and Signal Processing*, pages 497–500.

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.

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

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

L. Stark, 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*.

F. Wang, 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*.

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.

L. S. Zettlemoyer and M. Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proc. of Uncertainty in Artificial Intelligence*, pages 658–666.

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