USING HIDDEN MARKOV MODELS FOR TOPIC SEGMENTATION OF MEETING TRANSCRIPTS

Melissa Sherman[1], *Yang Liu*[2]

[1]Behavioral and Brain Sciences, University of Texas at Dallas, USA
[2]Computer Science Department, University of Texas at Dallas, USA

ABSTRACT

In this paper, we present a hidden Markov model (HMM) approach to segment meeting transcripts into topics. To learn the model, we use unsupervised learning to cluster the text segments obtained from topic boundary information. Using modified WinDiff and P_k metrics, we demonstrate that an HMM outperforms LCSeg, a state-of-the-art lexical chain based method for topic segmentation using the ICSI meeting corpus. We evaluate the effect of language model order, the number of hidden states, and the use of stop words. Our experimental results show that a unigram LM is better than a trigram LM, using too many hidden states degrades topic segmentation performance, and that removing the stop words from the transcripts does not improve segmentation performance.

Index Terms**—** Hidden Markov Model, Topic Segmentation, LCSeg, Meeting Transcript

1. INTRODUCTION

Topic segmentation aims to identify the story or topic boundary in a stream of text or speech. It is useful for many language processing tasks, such as summarization and language modeling. Segmenting broadcast news speech into stories has been part of the TDT (Topic Detection and Tracking) evaluation [1]. Various approaches have been developed for topic segmentation, most notably based on the lexical coherence of the text segments [2,3].

In this paper, we focus on the meeting recording genre. One challenge of separating a transcript of a meeting into topic segments is the nature of spoken speech. Meeting transcripts do not follow the rigid pattern of grammar and punctuation formatting associated with written text. There are many disfluencies in spontaneous speech. Topics are not clearly designated by paragraphs or key words as with written text and topic structure may not be cohesive. Therefore topic modeling using such transcripts must accommodate all of the extra noise generated in spoken speech. LCSeg [3], constructed with a lexical chain approach, has been successfully applied to topic segmentation in the meeting domain. In addition, machine learning based methods have been adopted for topic segmentation [4,5], using features from textual sources and speech related information.

While the HMM approach has been shown to be able to achieve good performance in the TDT broadcast news evaluation [6], there is significant difference between broadcast news and meeting domain genre. This study explores the effectiveness of the HMM for topic segmentation on meeting transcripts. Our experimental results have shown that HMMs yield better performance than LCSeg. We also investigate a few factors impacting the HMM segmentation approach.

2. RELATED WORK

Various unsupervised and supervised approaches have been proposed for topic segmentation for different genres. An early example of text segmentation using lexical cohesion is the TextTiling algorithm [2]. This unsupervised approach determines boundaries by finding the points where the lexical cohesion score is significantly different. The algorithm compares adjacent paragraphs in written articles to place topic boundaries. Good results have been reported using TextTiling for topic segmentation of scientific articles.

Another example of an unsupervised lexical cohesion algorithm is LCSeg [3]. It hypothesizes that a topic change in written text will occur at places where frequent repetition of words start and end, forming a chain. It gives a higher score to chains with more repeated terms, and also prefers shorter chains over longer ones. A cosine similarity measure decides the level of lexical cohesion between two lexical chains. LCSeg can perform topic segmentation with a known number of topic boundaries beforehand, or the algorithm can choose the number of topic boundaries by itself. Using the latter algorithm, LCSeg yielded a WinDiff segmentation error rate of 35% on a subset of the ICSI corpus [3].

Topic segmentation has been performed using machine learning techniques on AMI meeting transcripts [4,5]. [4] used a supervised classification approach to train decision trees to find topic boundaries. The classifier was trained with multiple features, including lexical cohesion, conversation features, and audio features such as overlapping speech, silence between speakers and surrounding cue phrases. The classifier was shown to perform better than LCSeg on the ICSI meeting corpus for top-level topic segmentation when the boundary numbers were not fixed in LCSeg.

[7] used Gibbs sampling to estimate the location of topic boundaries in the ICSI meeting transcripts. A variety of features were extracted to aid topic boundary detection, such as silence, speaker overlap, speaker activity and the median segment length. Better results were achieved when using extracted features than when no features were used.

HMMs have been proposed and evaluated previously for topic segmentation. The hidden states of the HMM represent generic background topics. [6] used an HMM in the TDT Pilot Study Corpus. In contrast to earlier experiments in [6], the structure of the HMM in [8] included a beginning state and an ending state. In [8], the language models in the hidden states were trained with a large collection of broadcast news text from the Linguistic Data Consortium. An additional step using decision trees was combined with the HMM to produce better segmentation results. HMMs have not been evaluated for topic segmentation in the meeting domain. Meeting transcripts are significantly different from written text or broadcast news transcription. Meeting transcripts do not have coherent topics, lack internal structure, and contain disfluencies or ill-formed sentences. The aim of our study is to evaluate whether we can develop an HMM for topic segmentation in the meeting domain.

3. APPROACHES

In this work, we use transcripts for topic segmentation without any acoustic/prosodic features. A sentence is used as the smallest unit of topic boundary selection, i.e., each sentence boundary is a topic change candidate.

3.1. HMM topic segmentation

In the HMM approach for topic segmentation, the hidden states represent some unknown "background topics." In a document, for the given sentence sequence *S*, the HMM aims to find the most likely topic sequence, T^*:

$$\begin{aligned} T^* &= \arg\max_T P(T \mid S) = \arg\max_T P(T)P(S \mid T) \\ &= \arg\max_{T_1 T_2 \cdots T_N} P(T_1)P(S_1 \mid T_1)\prod_{i=2}^{N} P(T_i \mid T_{i-1})P(S_i \mid T_i) \end{aligned} \quad (1)$$

The above formula illustrates a first order Markov model. Transition probabilities in the HMM capture the topic flow, while the emission probabilities represent the likelihood of the sentences generated by different topics. Transitions from one hidden topic to another indicate a topic change, and a self-looping transition indicates no change of topic.

In order to train the HMM, we need to estimate the model parameters of transition and observation probabilities. Since the transcripts are annotated with topic boundaries, but not topic labels, we choose to use unsupervised learning to find the "background topics." The reference topic boundaries were used to separate the transcripts into individual segments, and then the segments were automatically clustered into a pre-defined number of clusters, corresponding to the number of hidden states in HMM. For each hidden state, we train an n-gram language model (LM) using the text segments belonging to that cluster in order to estimate observation probabilities. The transition probabilities can be estimated from the training data once the cluster labels are known for each topic segment. However, due to the data sparsity, we choose not to learn the parameters in a supervised fashion. Similar to [8], we assign the transition probabilities heuristically: give a big probability to self-loop, and equally small probabilities to other state transitions. All of these parameters are optimized using a development set.

During the testing phase of HMM topic boundary detection, all sentences of a transcript are evaluated against the LMs associated with each hidden state to obtain the observation probabilities, $P(S_i|T_i)$. Then, we use the forward-backward algorithm to find the most probable hidden state, or "background topic," for each sentence in the transcript. Topic boundaries are placed between two sentences when there is a change of the best state hypothesis from one sentence to another. Note that during decoding, a weight is used to combine the transition and observation probabilities.

For unsupervised clustering of the text segments, we used CLUTO [10] to create the pre-defined number of clusters, based on the objective function to minimize the inter-cluster similarity and maximize the intra-cluster similarity. The SRILM toolkit [9] was used to create the n-gram LMs for the observation probabilities of each hidden topic and implement the HMM decoding.

3.2. HMM without stop words

Stop words are words that occur frequently in a given corpus. The inverse document frequency (IDF) of a word can be used to represent whether it is topic indicative. The IDF for a word is often computed as $\log(N/N_i)$, where N is the number of documents in a collection and N_i is number of documents containing this word. We consider a word as a stop word if its IDF is lower than a defined threshold. We expect that stop words are not topic indicative and contain little relevant information to determine what topic the sentence is from. To see if we can obtain better performance with less noise produced by the stop words, we remove a varying number of stop words from the meeting transcripts and then apply the HMMs as described previously.

4. EXPERIMENTS

4.1. Experimental setup

We used the ICSI meeting corpus [11], which contains 75 recordings from freeform meetings. Each meeting is approximately an hour long and was transcribed manually, with annotated dialogue acts [12]. Our testing set includes 25 ICSI meetings, chosen as a representative subset of the whole corpus, with the remaining 50 meetings as the training/development set. The topic boundary annotation used in this study is the AMI topic annotation for the ICSI meeting corpus [15]. The average number of topic boundaries per meeting is 5.6. We use the dialog act annotation in the corpus to form the "sentences" mentioned in the HMM approach. Topic boundaries can only be placed at the beginning or the end of a dialog act.

4.2. Evaluation metrics

To evaluate topic segmentation performance, we created a new variation of the WinDiff (W_d) [13] and P_k [14] metrics. Both original metrics observe a scrolling window of text in the document and compare the number of segmentation boundaries in the reference and the system hypothesis. If they are different, a penalty is incurred in W_d, in other words, the W_d score is the probability that the numbers of hypothesis and reference boundaries are different in any given window throughout the text. For the P_k metric, the presence of a topic change in one document but not the other incurs a penalty. This results in the probability that any two sentences, a window distance apart, are incorrectly listed as being in the same topic. For both metrics, a lower score indicates better performance.

During our development phase, we noticed some inconsistencies in the results produced by P_k and W_d — hypothesis boundaries placed within the window size of the beginning or end of the document yielded less of an error than boundaries placed in the middle of the hypothesis document. To correct for this under-penalty, the window was placed so that every possible boundary point was observed K times, with K as the window size. Due to more opportunities for errors to be counted, these "extended window" version of the metrics yield higher error rate than the traditional P_k and W_d metrics, but we feel the metrics reflect a more accurate evaluation of the hypothesized boundaries. To differentiate our "extended window" metrics from the original P_k and W_d metrics, we will use the abbreviations: eP_k and eW_d.

4.3. Results

4.3.1. Baseline results

The baseline system was created by randomly placing the same number of boundaries as in the reference label into a test document.[1] Table 1 shows the average eP_k and eW_d results using the 25 test meetings. Each set of boundaries was randomly placed 100 times, and the average value over the test set of meetings was reported. For comparison, we also show results for both 0 and 1 hypothesis boundary per document using the same process as above.

Table 1 clearly shows the preference for fewer boundaries per document using both the eP_k and eW_d metrics. Even though placing no boundaries receives a better score, it is not a useful segmentation. Therefore, in the following experiments, in addition to using eP_k and eW_d metrics, we also provide the average boundary numbers, in order to avoid the bias caused by fewer boundaries (and thus better scores).

Table 1. Baseline results

# of boundaries	eP_k	eW_d
0	0.381	0.381
1	0.409	0.409
Same as reference (avg 5.96)	0.495	0.523

4.3.2. LCSeg Results

We ran LCSeg on the 25 test meeting transcripts. The first trial allowed LCSeg to choose the number of boundaries to place, while the second trial forced LCSeg to place the same number of boundaries as the reference transcript. Table 2 shows the results of those trials, with each value representing an average of the 25 test meetings. As expected, there is an improvement when LCSeg is provided with the correct number of topic boundaries.

Table 2. LCSeg results

LCSeg boundaries	Avg # of boundaries	eP_k	eW_d
System choice	12.84	0.425	0.574
Same as reference	5.96	0.352	0.422

4.3.3. HMM results

For the HMM approach, we investigate the impact of the following factors: using different LM orders (unigram vs. trigram) for the observation probabilities, different number of hidden states (number of clusters), and different data size for model training. Each language model was trained using Good-Turing discounting.

To obtain parameters of the transition probabilities and interpolation weight, we used cross validation on the 50 training meetings. We performed 10-fold cross validation (with 5 meetings in each subset) to obtain the best values for the unigram and trigram LMs. When choosing the optimal values, we looked at the eP_k and eW_d scores, and the number of system hypothesized boundaries. This avoided combinations that performed well due to a low number of hypothesis boundaries. The final log transition likelihood and the weight for the unigram models were -6 and 0.3, while the trigram models used the values of -8 and 0.5 for their tests. Using these parameters, we trained the model on either 10, 30 or 50 meetings and then tested the model on the 25 test meetings. Tables 3 shows the HMM topic segmentation results using unaltered transcripts.

Table 3. HMM results

# of States	LM order	Training Meetings	eP_k	eW_d	Avg # of boundaries
5	1	10	0.332	0.404	4.44
		30	0.327	0.398	4.60
		50	0.335	0.408	4.96
	3	10	0.375	0.406	3.28
		30	0.364	0.405	3.64
		50	0.372	0.416	4.32
10	1	10	0.331	0.410	5.40
		30	0.345	0.418	5.72
		50	0.336	0.407	5.04
	3	10	0.359	0.446	6.80
		30	0.374	0.455	6.56
		50	0.359	0.426	5.60
20	1	10	0.400	0.515	9.44
		30	0.355	0.457	8.20
		50	0.354	0.433	6.00
	3	10	0.451	0.606	13.60
		30	0.415	0.529	10.68
		50	0.372	0.445	6.16

We can see that when using a larger number of hidden states (e.g., 20), an increase in the amount of training data reduces the error rate. However, for the models with 5 or 10 states, they are able to perform well even if they are only trained on few meetings. The average number of system boundaries shows different patterns depending on the number of hidden states in the model. For the 5 hidden state model, in both unigram and trigram LMs, the average number of boundaries increases as the amount of training data increases. In the 20 hidden state models, an increase in training data decreases the average number of system boundaries. The results in Table 3 also show that a unigram LM consistently yields better segmentation performance than a trigram LM for the generation of observation probabilities in this HMM task. This is possibly due to the data sparsity in trigram LM training. In addition, since this problem setup is similar to a topic classification task, it is not surprising that the unigram bag-of-words approach outperforms the trigram model as most previous work has shown.

Compared to the baseline and LCSeg results, we find that the HMM outperforms the baseline (Table 1). In general, HMM yields a lower error than LCSeg (Table 2) when the number of reference topic boundaries is unknown to LCSeg. For some configurations, the HMM can also outperform LCSeg even if the number of boundaries is known to LCSeg. Even though we cannot directly compare the results of the HMM to the supervised results in [5], the HMM shows a similar pattern of improved performance over the LCSeg results with an unknown number of boundaries.

Since HMM and LCSeg are two different approaches, we expected they make different errors and their combination may help system performance. We explored a method to first convert the HMM system hypothesis to posterior probabilities and combine them at the decision level, however, this combination did not yield any increase in performance.

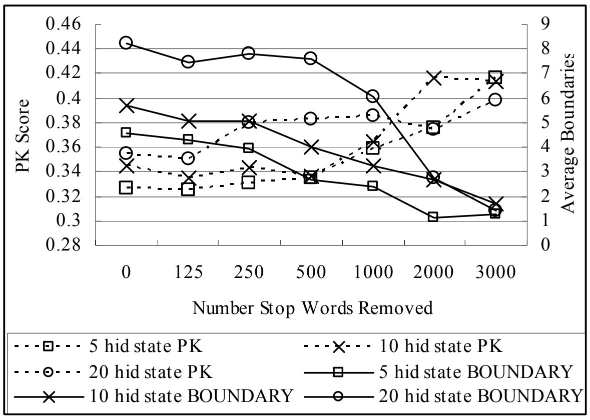
4.3.4. Impact of stop words

In order to place more weight on the content words of a document, we investigated removing stop words from the transcripts in the training and testing phases of the HMM

[1] Note that this is a rather high baseline since we are using the reference number of topic boundaries.

framework. The IDF values for the words in all documents were computed using the complete set of ICSI meeting transcripts as “documents.” We removed words with a low IDF from the transcripts and applied HMM on the newly created transcripts. Only unigram LMs were used in this experiment because stop words had been removed from the transcripts, which destroyed the original word context required for trigram LMs. Figure 1 shows the topic segmentation results by varying the number of stop words removed from each transcript. All of the models were trained on 30 meetings. The left Y-axis is the eP_k score, while the right Y-axis shows the average number of boundaries. Because of the similar patters of eP_k and eW_d , we only show the results for eP_k here.

Figure 1. Performance of HMMs with stop words removed from the transcripts.



As we can see from the graph, the average number of boundaries placed decreases as the number of stop words removed from the transcript increases. Removing any number of stop words does not significantly increase the performance for any of the systems. For all of the hidden states, the performance degrades as the number of stop words removed increases further. The lower error rate along with a decrease in the average amount of boundaries is a strong indication that the systems are not able to place accurate boundaries as the number of stop words removed increases.

We also performed similar experiments using 10 and 50 training meetings. When the number of training meeting was small, removing too many stop words (e.g., 3000) resulted in zero boundaries for all number of hidden states. This indicated that too much content was removed from the transcripts, and the HMM model could not differentiate between topics. Our stop word removal results suggest that using all of the words in the meeting transcripts is more effective for topic boundary detection.

5. CONCLUSION

In this paper, we have demonstrated the effectiveness of HMMs for topic segmentation using transcripts of freeform meetings. Our experiments on the ICSI meeting corpus show that the HMM yields better performance than LCSeg, a state-of-the-art lexical cohesion based unsupervised approach. We have investigated a few factors that impact the HMM performance, including the LM order for the observation probabilities, number of hidden states, training data size, and the removal of stop words. We find that a unigram LM outperforms a trigram LM and the best performance is achieved when using fewer hidden states. Having a larger training data set helps more for a larger number of hidden states (e.g., 20) than for smaller numbers of hidden states (5 or 10.) Removing stop words does not yield any increase in performance for any model configurations.

This study used human transcripts and annotated dialog acts as sentences. In our future work, we will evaluate the HMM using speech recognition output and automatic sentence segmentation. In addition, we will incorporate acoustic/prosodic features in this framework. Finally, we plan to evaluate the impact of topic segmentation on downstream language processing tasks, such as summarization and keyword extraction.

6. ACKNOWLEDGEMENT

We thank Michel Galley for sharing the topic annotation of the ICSI meeting corpus and the LCSeg program. This work is supported by NSF award IIS-0714132. Any opinions expressed in this work are those of the authors and do not necessarily reflect the views of NSF.

7. REFERENCES

[1] Online: http://www.nist.gov/speech/tests/tdt/

[2] Hearst, M. A., “Multi-Paragraph Segmentation of Expository Text”, In Proceedings of ACL, 1994.

[3] Galley, M., McKeown, K., Fosler-Lussier, E., and Jing, H., “Discourse Segmentation of Multi-Party Conversation”, In Proceedings of ACL, 2003.

[4] Online: http://corpus.amiproject.org/

[5] Hsueh, P. and Moore, J., “Automatic Topic Segmentation and Labeling in Multiparty Dialogue”, In Proceedings of the first IEEE/ACM workshop on Spoken Language Technology, Aruba, 2006.

[6] Carp, I., Gillick, L., Lowe, S., Van Mulbregt, P., and Yamron, J.P., “A hidden Markov Model Approach to Text Segmentation and Event Tracking”, In Proceedings of ICASSP, 1333-1336, 1998.

[7] Dowman, M., Griffiths, T., Koerding, K., Purver, M. Savova, V. and Tenenbaum, J. “A Probabilistic Model of Meetings that Combines Words and Discourse Features”, *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 7, 2008.

[8] Hakkani-Tur, D., Stolcke, A., Shriberg, E., and Tur, G., “Integrating Prosodic and Lexical Cues for Automatic Topic Segmentation”, *Computational Linguistics*, 27, 31-57, 2001.

[9] Stolcke, A., “SRILM - An Extensible Language Modeling Toolkit”, In Proceedings of ICSLP, 2002.

[10] Online: http://glaros.dtc.umn.edu/gkhome/views/cluto/

[11] Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., and Wooters, C., “The ICSI Meeting Corpus”, In Proceedings of ICASSP, 2003.

[12] Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., and Carvey, H., “The ICSI Meeting Recorder Dialog Act (MRDA) Corpus”, In Proceedings of 5th SIGDial Workshop, 97-100, 2004.

[13] Hearst, M.A. and Pevzner, L., “A Critique and Improvement of An Evaluation Metric for Text Segmentation”, *Computational Linguistics*. *28*, 19-36, 2002.

[14] Beeferman, D., Berger, A. and Lafferty, J. “Statistical Models of Text Segmentation”, *Machine Learning*, 34 (1-3), February 1999.

[15] Murray, G, Renals, S, and Carletta, J. “Extractive Summarization of Meeting Recordings”, In Proceedings of Interspeech, 2005.