Topic Segmentation with an Aspect Hidden Markov Model

David M. Blei[*]
University of California, Berkeley
Dept. of Computer Science
495 Soda Hall
Berkeley, CA, 94720, USA
blei@cs.berkeley.edu

Pedro J. Moreno
Compaq Computer Corporation
Cambridge Research Laboratory
One Cambridge Center
Cambridge, MA, 02142, USA
Pedro.Moreno@compaq.com

ABSTRACT

We present a novel probabilistic method for topic segmentation on unstructured text. One previous approach to this problem utilizes the hidden Markov model (HMM) method for probabilistically modeling sequence data [6]. The HMM treats a document as mutually independent sets of words generated by a latent topic variable in a time series. We extend this idea by embedding Hofmann's aspect model for text[5] into the segmenting HMM to form an aspect HMM (AHMM). In doing so, we provide an intuitive topical dependency between words and a cohesive segmentation model. We apply this method to segment unbroken streams of New York Times articles as well as noisy transcripts of radio programs on SpeechBot [1], an online audio archive indexed by an automatic speech recognition engine. We provide experimental comparisons which show that the AHMM outperforms the HMM for this task.

Topics: Machine Learning for IR; Topic Detection and Tracking

1. INTRODUCTION

In the classical information retrieval (IR) problem, a user searches in a corpus of text for documents which satisfy her information needs. This framework assumes a notion of document i.e. that the corpus is divided into cohesive sets of words each of which expresses particular information.

In some search-worthy text corpora, such as newswire feeds, television closed captions, or automatic speech recognition (ASR) transcripts of streaming audio, there is no explicit representation of a document. There are implicit document breaks (e.g. television shows, radio segments) but no clear demarcations of where they occur. Segmentation is a critical subtask of the IR problem in these corpora.

To this end, we implemented a novel probabilistic method of topic segmentation which combines a segmenting hidden Markov model [6] and an aspect model [5]. In this paper, we describe our method and demonstrate good results when applied to noisy ASR transcripts and streams of clean (error-free) unsegmented text.

This paper is divided into six sections. In section 2, we summarize of previous techniques and describe how our method relates to them. In section 3, we describe the standard HMM segmentation approach. In section 4, we describe the theory behind the aspect HMM approach. In section 5, we report on experiments on both clean and ASR text. In section 6, we present our conclusions and suggestions for future work.

2. PREVIOUS WORK

There is a considerable body of previous research on which this work builds. Hearst [4] developed the *TextTiling* algorithm which uses a word similarity measure between sentences to find the point between paragraphs at which the topic changes. This approach is effective on clean text with explicit sentence and paragraph structure. However, it is difficult to implement on text produced by a speech recognition engine. In addition to the unstructured nature of ASR output, speech recognition engines on unrestricted audio often have word error rates in the range of 20% to 50%. Since Hearst's algorithm computes cosine similarity between relatively small groups of words on either side of a sentence boundary, it is unclear whether it is robust in the face of many erroneous words.

Beeferman et al. [1] introduced a feature-based probabilistic segmentation method which does not require text with

[*] This work was done during a summer internship at the Cambridge Research Lab.

[1] A public web site available at `http://www.speechbot.com`

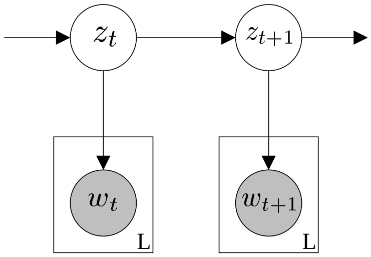

Figure 1: A graphical model representing the segmenting HMM. Circles represent random variables and arrows indicate possibly dependency. The plate around w_t denotes that this random variable is repeated L times for each topic variable in the series.

paragraph and sentence structure. Though their method works well, many of the derived features are based on identifying cue-words which indicate an impending topic shift. In our domain, high error rates often cloud such cue words making them difficult to learn and detect.

The method we present builds directly on the Hidden markov model (HMM) approach of Mulbregt et al. [6]. We extend this model by embedding the aspect model [5] in the HMM. This gives rise to a unified model within which we find both segment clusters to train transition probabilities and language models to determine observation emission probabilities.

3. HMM SEGMENTATION

In the segmenting HMM framework, an unsegmented document is treated as a collection of mutually independent sets of words. The model posits that each set is probabilistically generated by a hidden topic variable in a series. Transition probabilities between topics determine the next hidden variable in the sequence.

As a generative model, the HMM posits that a document is produced by the following process. First, choose a topic from an initial distribution of topics. Then, generate a set of L independent words from a distribution over words associated with that topic. Finally, choose another topic, possibly the same topic, from a distribution of allowed transitions and repeat this process. Given a new, unsegmented document, one inverts this process by calculating the most likely set of topics which generated the L-word sets of the given document. Topic breaks occur at the points where the value of the topic variables change.

More formally, $o_t = \{w_{t,1}, w_{t,2}, w_{t,3}, \ldots, w_{t,L}\}$ are sets of L words and are generated by a topic z_t. Each z_t depends only on z_{t-1} and the o_t are independent of each other given z_t. This is illustrated in the graphical model in figure 1.

The HMM is parameterized by a transition probability distribution between topics and a set of topic-based unigram language models $P(w|z)$ for each possible value of z. To train the model, a set of segments from a corpus is clustered using the k-means algorithm. A unigram language model is computed for each of these clusters and an appropriate smoothing technique is applied to account for sparsity. The transition probability distribution between topic states $P(z_{t+1}|z_t)$ is a parameter which is separately tuned in [6]. We simply use normalized counts of transitions between clusters in the training set to estimate it. Note that this model requires a segmented corpus to train, but works in an unsupervised manner to cluster those segments.

To segment a new document, the stream of text is divided into a sequence of observations o_t of L words each. The Viterbi algorithm [7], a dynamic programming technique, is used to find the most likely hidden sequence of topic states $Z = \{z_0, z_1, \ldots, z_T\}$ given an observed sequence of word sets $O = \{o_0, o_1, \ldots, o_T\}$. Topic breaks occur when $z_t \neq z_{t+1}$.

This model is an effective segmentation framework on both clean and ASR text. However, it suffers from the naive Bayes assumption that the words within each observation are mutually independent given a topic.

$$P(o_t|z) = \prod_{i=1}^{L} P(w_i|z)$$

As L gets large, this assumption works well for computing $P(o_t|z)$. However, the larger L becomes, the less precise the resulting segmentation will be since the model can only hypothesize topic breaks between sets of words. The window (i.e. L) must be large enough to give an accurate estimate of $P(o|z)$ while small enough to detect a segmentation point with good granularity.

4. ASPECT HMM SEGMENTATION

A segmenting aspect HMM (AHMM) is a hidden Markov model in which each hidden state is an instance of the latent variable in an embedded aspect model. This aspect model determines both the observation emission probabilities and training segment clusters to find the transition probabilities. As in the segmenting HMM, each observation is a set of L words and we use the Viterbi algorithm to find topic breaks.

4.1 The aspect model for documents and words

In this section we summarize Hofmann's aspect model as it applies to text. For a detailed discussion, see [5].

The aspect model is a family of probability distributions over a pair of discrete random variables. In text data, this pair consists of a document label and a word. It is important to understand that in the aspect model, a document is not represented as the set of its words but simply a label which identifies it. It is associated with its corresponding set of words through each document-word pair.

This model posits that the occurrence of a document and a word are independent of each other given a topic or factor. Let d denote a segment from a presegmented corpus, w denote a word, and z denote a topic. Under this independence assumption, the joint probability of generating a particular topic, word, and segment label is

$$P(d, w, z) = P(d|z)P(w|z)P(z).$$

The $P(w|z)$ parameter is a language model conditioned on the hidden factor. The $P(d|z)$ parameter is a probability distribution over the training segment labels. The $P(z)$ distribution is a the prior distribution on the hidden factor.

Given a corpus of N segments and the words within those segments, the training data for an aspect model is the set of pairs $\{(d_n, w_n^d)\}$ for each segment label and each word in those segments. We can use the Expectation Maximization (EM) algorithm [2] to fit the parameters from an uncategorized corpus. This corresponds to learning the underlying topics of a corpus $P(w|z)$ as well as the degree to which each training document is about those topics $P(d|z)$.

In the E-step, we compute the posterior probability of the hidden variable given our current model. In the M-step, we maximize the log likelihood of the training data with respect to the parameters $P(z)$, $P(d|z)$, and $P(w|z)$. The E-step is

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')}$$

The M-step is

$$\begin{aligned} P(d|z) &= \frac{\sum_{w\in W} P(z|d,w)n(d,w)}{\sum_{w\in W}\sum_{d'\in D} P(z|d',w)n(d',w)} \\ P(w|z) &= \frac{\sum_{d\in D} P(z|d,w)n(d,w)}{\sum_{w'\in W}\sum_{d\in D} P(z|d,w')n(d,w')} \\ P(z) &= \frac{\sum_{d\in D}\sum_{w\in W} P(z|d,w)n(d,w)}{\sum_{z'}\sum_{w\in W}\sum_{d\in D} P(z'|d,w')n(d,w)} \end{aligned}$$

where $n(d,w)$ is the number of times word w appears in document d.

Since d refers to a training document label, the number of parameters of $P(d|z)$ grows linearly with the size of the training data making the aspect model quite prone to overfitting. To avoid this, we use tempered EM as described in [5]. Essentially, we hold out a portion of our training data for cross validation purposes after the E-step. When the performance decreases on the hold-out data, we reduce a parameter $\beta \leq 1$ which tempers the effect of the next M-step on the parameters of the model. In the case of a segmenting AHMM, we cross validate by checking the segmentation accuracy on a held out set of transcripts (see section 5.3 for a description of the error measure). We stop training when reducing β no longer improves performance on the segmentation of the hold-out training data.

4.2 The aspect HMM

The segmenting AHMM is an HMM for which the hidden topic state is the z random variable in a trained aspect model. This is depicted in figure 2. The AHMM works in exactly the same way as the HMM except that the words from the selected hidden factor are generated via the aspect model rather than independently generated.

To train an AHMM, we train an aspect model on a set of training segments as described in section 4.1. We cluster the training segments by the $P(d|z)$ parameter.

$$\text{cluster}(d) = \arg\max_i P(d|z_i)$$

Finally, we compute transition probabilities between clusters and initial probabilities of each cluster.

Note that the aspect model does not represent clusters in the way that we compute them. Each d is represented by $P(d|z)$, a probability for each latent factor. There is no theoretical reason that the factor with maximum probability should indicate a cluster assignment. However, in practice, $P(d|z)$ for a fixed d is peaked towards one value of z. In this case, we feel justified in assigning each segment to the factor with maximal probability.

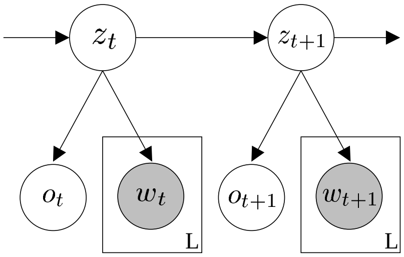

Figure 2: A graphical model representing a segmenting AHMM

The AHMM segments a new document by dividing its words into observation windows of size L and running the Viterbi algorithm to find the most likely sequence of hidden topics which generated the given document. Segmentation breaks occur when the value of the topic variable changes from one window to the next. The Viterbi algorithm requires the observation probability $P(o_t|z)$ for each time step. While the HMM uses the naive Bayes assumption to compute this distribution, we treat each o_t as a new segment label and compute $P(o_t|z)$ via the aspect model.

One problem with the aspect model is that it is not a truly generative model with respect to document labels. As we mentioned in section 4.1, the $P(d|z)$ parameter is a discrete distribution over the set of *training* documents. Therefore, the model can only compute conditional probabilities about those segments which it was exposed to in training. In the Viterbi algorithm, we need to find $P(o_t|z)$ for some observation window o_t. This observation is *not* a document label that the model has seen before. To properly find $P(o_t|z)$, one should retrain the model using EM on the training corpus as well as o_t and the words it contains. However, this is very inefficient. In practice, one can use an online approximation to EM to find $P(o_t|z)$. We use a variant as described in [3].

Let $o_{t,i} = \{\epsilon, w_{t,1}, w_{t,2}, \ldots, w_{t,i}\}$ where $w_{t,0} = \epsilon$ denotes no word and $o_{t,L} = o_t$ denotes the full observation. We approximate $P(z|o_t)$ recursively as follows.

$$\begin{aligned} P(z|o_{t,0}) &= P(z) \\ P(z|o_{t,i+1}) &= \frac{1}{i+1}\frac{P(w_{i+1}|z,o_{t,i})P(z|o_{t,i})}{\sum_{z'} P(w_{i+1}|z')P(z'|o_{t,i})} + \\ &\quad \frac{i}{i+1}P(z|o_{t,i}) \end{aligned}$$

Then we use Bayes rule to find $P(o_t|z)$.

$$P(o_t|z) = \frac{P(z|o_t)P(o_t)}{P(z)}$$

Note that $P(o_t)$ is not a meaningful probability. However, the Viterbi algorithm only needs to compute $P(o_t|z)$ for a single observation at a time. Thus, $P(o_t)$ behaves like a scaling constant and we can compute $P(o_t|z)$ up to this factor. Finally, since the Viterbi algorithm only compares probabilities, we can use this proportional probability without any loss.

These formulae reflect an online approximation of one E-step in the EM algorithm. We present here an intuitive derivation to illustrate why they make sense as such an approximation. We would like to recursively estimate $P(z|o_t)$ from partial estimates of $P(z|o_{t,i})$. First, notice that $o_{t,0}$ is the empty word. This immediately gives us the base case.

$$P(z|o_{t,0}) = P(z)$$

We can express $P(z|o_{t,i})$ in terms of our previous information as follows.

$$P(z|o_{t,i}) = \sum_{w \in o_{t,i}} P(w)P(z|w, o_{t,i-1})$$

We assume that, in a partial observation sequence o_i, the marginal probability of selecting any word is simply $1/(i+1)$. Observe that when $w \neq w_i$, the word is assumed to have been accounted for in $P(z|o_{i-1})$ and is absorbed in the conditioning. When $w = w_i$, we can compute $P(z|w_i, o_{i-1})$ by a simple application of Bayes rule.

$$\begin{aligned} P(z|o_{t,i}) &= \frac{1}{i+1}P(z|w_i, o_{t,i-1}) + \frac{i}{i+1}P(z|o_{t,i-1}) \\ &= \frac{1}{i+1}\frac{P(w_i|z, o_{t,i-1})P(z|o_{t,i-1})}{P(w_i)} + \\ &\quad \frac{i}{i+1}P(z|o_{t,i-1}) \\ &= \frac{1}{i+1}\frac{P(w_i|z)P(z|o_{t,i-1})}{\sum_{z'} P(w_i|z')P(z'|o_{t,i-1})} + \\ &\quad \frac{i}{i+1}P(z|o_{t,i-1}) \end{aligned}$$

The final equation expresses $P(z|o_{t,i})$ in terms of $P(z|o_{t,i-1})$. As the approximator sees more words in a single observation, it refines its posterior distribution of the topic. It uses this refined posterior to weight the distribution of the next word.

5. EXPERIMENTAL RESULTS

We applied this segmentation model to two large corpora. First, we examined SPEECHBOT transcripts from *All Things Considered* (ATC), a daily news program on National Public Radio. Our corpus spans 317 shows from August 1998 through December 1999. Within these shows there are 4,917 segments with a vocabulary of 35,777 unique terms. The shows constitute about 4 million words. We estimated the word error rate in this corpora to be in the 30% to 40% range. Note that these are only estimates computed from sampling the corpora as perfect transcripts are unavailable to us.

Additionally, we analyzed a corpus of 3,830 articles from the *New York Times* (NYT) to compare the ASR performance with error-free text. This corpus constitutes about 4 million words with a vocabulary of 70,792 unique terms. In all reported experiments, we learn an aspect model with 20 hidden factors.

5.1 Aspect model EM training

Figure 3 illustrates the performance on held out data during the tempered EM training of the aspect model (see section 4.1). Though the NYT corpus takes longer to converge (due to the higher vocabulary size), it learns more quickly than the ATC corpus since the text contains no errors. The ATC converges faster (due to the smaller vocabulary size) but stays at a lower success rate (see section 5.3 for how we measure success) for several iterations before performance improves.

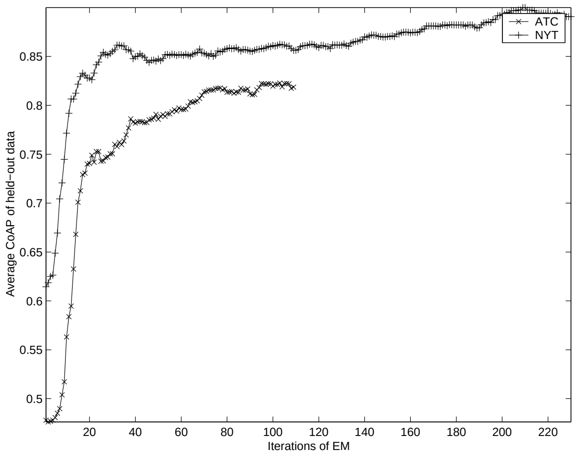

Figure 3: Tempered EM convergence in the ATC and NYT corpora

5.2 Sample results and topic labels

In our experiments, we used three variants of our two corpora. First, we created random sequences of segments from the ATC corpus. Second, we created random sequences from the NYT corpus to compare clean versus noisy segmentation. Finally, we used the actual aired sequences of ATC segments since this is domain of the primary problem which we are trying to tackle.

In the random sequences of segments, we attained almost perfect segmentation on both corpora. However, the results are mixed with the original broadcasts of the ATC. Figure 4 shows a segmentation from a correctly sequenced transcript of ATC on April 29, 1999. The segmentation is not perfect but hypothesizes the detected topic breaks at approximately the correct points in the program. At first, there seem to be many missed breaks. We argue however that these missed story breaks do not always constitute topic breaks and therefore are not indicative of the performance of our model. To illustrate this, we explore a method of topic labeling based on the language model parameters of the aspect model.

One way of identifying the topics which the segmenter finds is by the top fifteen words of the $P(w|z)$ parameter for the value of z which the Viterbi algorithm assigned to a particular segment. Figure 5 lists these word sets (denoted by a letter) as they correspond to the topics in the segmentation (denoted by a number). For example, story 14 is about the Israeli/Palestinian conflict. Its corresponding segment in the hypothesis segmentation can be described by the words in topic **F** which include `peace`, `israeli`, and `palestinian`.

Analysis of this correspondence often explains missed topic breaks. Articles 11 and 12 are both about the Kosovar refugees. Understandably, they are both assigned to topic **A** and the break between stories goes undetected.

Note that the segmenter can work even if the top words of $P(w|z)$ fail to give a good topic description. The story

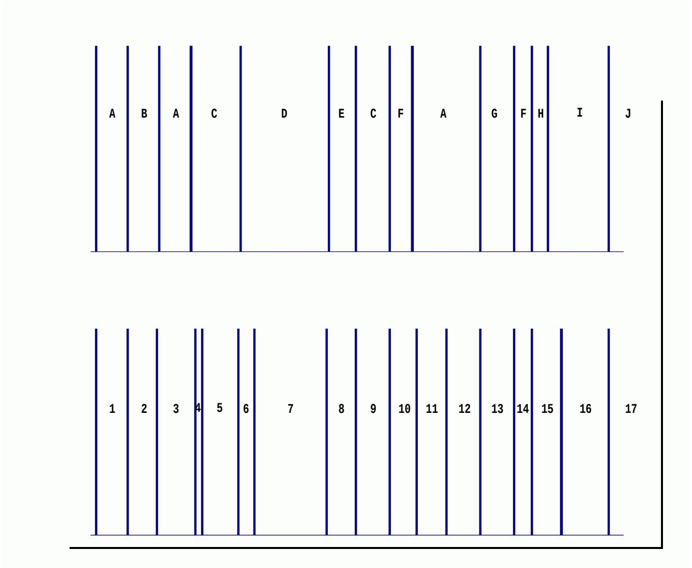

Figure 4: A segmentation of *All Things Considered* from April 29, 1999. The top diagram is the hypothesis segmentation. The bottom diagram is the true segmentation.

about deformed frogs is assigned topic **I**, a rather generic language model with no real descriptive words. However, the subsequent story about the economy fits topic **J** so well that the AHMM is able to properly detect the break.

5.3 Quantitative Results

We use the *co-occurrence agreement probability* (CoAP) introduced in [1] to quantitatively evaluate our segmenter. The CoAP is defined as

$$P(\text{agreement}) = \sum_{(i,j)} D(i,j)\delta_R(i,j) \oplus \delta_H(i,j)$$

The function $D(i,j)$ is a probability distribution over the distances between words in a document; the δ functions are 1 if the two words fall in the same segment and 0 otherwise; and $\oplus$ function indicates agreement between the operands.

In our case, $D(i,j) = 1$ if the words are k words apart and 0 otherwise. With this choice of D, the CoAP is a measure of how often a segmentation is correct with respect to two words that are k words apart in the document. Following [1], we choose k to be half the average length of a segment in the training corpus, 170 in the ATC corpus, and 200 in the NYT corpus.

A useful interpretation of the CoAP is through its compliment [1]

$$P(\text{disagreement}) = P(\text{missed})P(\text{seg}) + (1 - P(\text{seg}))P(\text{false})$$

where $P(\text{seg})$ is the a priori probability of a segment, $P(\text{missed})$ is the probability of missing a segment, and $P(\text{false})$ is the probability of hypothesizing a segment where there is no segment.

Figure 6 shows the error and its decomposition for three experiments: the NYT corpus with randomly generated sequences of articles; the ATC corpus with randomly generated sequences of segments; and the ATC corpus with the true ordering of segments as they were aired. It is interesting to note that our system tends to undersegment as indicated by the high $P(\text{missed})$. Furthermore, in the actual ATC or-

A nato, military, kosovo, said, air, get, today, forces, troops, people, refugees, says, yugoslav, re, to, war

B president, house, republican, replublicans, clinton, senate, impeachment, democrats, said, think, get, white, today, people, congress

C school, students, schools, get, know, think, says, people, good, like, two, just, children, year, education

D get, know, like, good, new, re, just, two, people, time, says, think, music, see

E says, get, health, people, care, new, two, women, years, re, year, patients, good, medical, study

F nato, president, peace, israeli, israel, minister, palestinian, today, said, get, agreement, prime, kosovo, war, milosevic

G olympic, two, said, new, information, today, good, committee, people, nineteen, time, year, internet

H people, get, says, said, think, two, good, new, president, today, time, year, nineteen, years

I get, think, people, know, just, re, says, time, good like, two, don, new, things, say, see, going

J today, said, two, get, president, says, market, economy, good, government, new, economic, year, percent, time, hundred

1. NPR's Julie McCarthy reports from NATO headquarters in Brussels on the status of the air war over Yugoslavia including a missile that went astray and landed near Sophia the capital of Bulgaria.
2. A new NPR Kaiser Kennedy School Poll released today shows substantial support for current US actions in Yugoslavia.
3. Congress is divided in its sentiments about the war in Kosovo.
4. Linda updates the news from Littleton Colorado where another funeral was held today and the investigation continues into the planning of the attack on Columbine High School.
5. Linda and Noah read letters from All Things Considered listeners.
6. New York City teens react to the Littleton Colorado high school tragedy.
7. Today marks the centennial of the birth of Edward Kennedy Ellington.
8. Government figures indicate teenage pregnancy has fallen sharply reducing the countrys overall birth rate.
9. The Florida legislature is expected Thursday to adopt the nations first statewide school voucher program.
10. NPRs Tom Gjelten reports that former Russian Prime Minister Viktor Chernomyrdin has undertaken a twoday diplomatic mission aimed at restoring peace in Yugoslavia.
11. Sarah Chayes reports from Tirana Albania on families that have taken in Kosovar refugees.
12. Barbara Mantel reports on the beginning of efforts to bring some Kosovar refugees to the U.S temporarily.
13. NPRs Mike Shuster reports that a scientist who was fired from his job at the Los Alamos National Laboratory on suspicion that hed transferred U.S weapons secrets to China may have caused more damage than previously thought.
14. NPR senior news analyst Daniel Schorr says that in the midst of the crisis in Kosovo the ageold Israeli/Palestinian conflictfor nowstill has a chance for a peaceful settlement.
15. NPRs Wade Goodwyn reports funeral services were held today for yearold Isaiah Shoels. Shoels was a football player and the only black student killed in the Columbine High massacre.
16. NPRs Richard Harris reports that scientists have discovered why some North American frogs have been suffering from disturbing deformities such as extra legs or missing legs.
17. NPRs Jim Zarroli reports on Wall Streets prediction that the millennium weekend will pass without significant bugs for stock exchanges or major brokerages.

Figure 5: Summary words (up) and ground truth summaries (down) from the ATC segment in figure 4

Source	P(missed)	P(false)	P(disagree)
Random NYT	0.123	0.080	0.096
Random ATC	0.263	0.052	0.143
Actual ATC	0.434	0.063	0.233

Figure 6: CoAP results on the ATC and NYT corpora. In the case of randomly generated transcripts, the reported results are the mean over ten sets of random transcripts taken from the same set of testing segments.

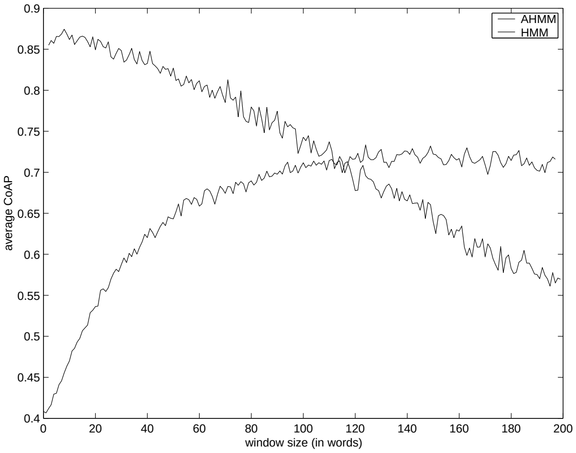

Figure 7: Window width vs. CoAP for the HMM and AHMM in the NYT corpus

derings $P(\text{missed})$ is even higher due to the phenomenon of multiple segments with similar topics (see section 5.2).

Figure 7 is a comparison between the AHMM and HMM over window widths from 2 to 200. AHMM segmentation outperforms HMM segmentation for small window widths. However, as we increase the window size, the performance of the aspect model decreases. This is due to two facts. First, the precision of the segmenter decreases, causing a slight decrease in score. More importantly however, this behavior occurs because we are using an *approximation* of $P(o_t|z)$. In the approximation scheme described in section 4.2, words in the beginning of the window are weighted more heavily than words towards the end of the window. Therefore, as the window size increases, more words make less impact on the observation distribution and the segmenter does not perform as well.

The HMM does well on large windows since all words are counted equally. However, this increase in performance is at the expense of low segmentation granularity. While the HMM performs better than the AHMM for large windows, it never attains the performance of the AHMM in small windows. Typically, the AHMM reaches peak performance at a window size of 10-15 words. The HMM begins to perform better than the AHMM at around 100 words.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a new approach to text segmentation using a unique probabilistic model that combines an aspect model with an HMM. This is a unified framework within which we learn both document clusters for training and observation probabilities for new segmentations. The AHMM does well with small windows of words allowing for a more precise segmentation than with the HMM.

We have experimented with this system on noisy text sources produced by a speech recognition system. Since our model does not use syntactic structure information, we can segment this output and accurately hypothesize topic transition points. Our results on transcripts produced by the SpeechBot system are quite encouraging.

Future work in this area has several directions. First, we would like to incorporate segmentation into the Speechbot IR framework in a principled way and measure its success. Second, we would like to use the topic labels to categorize the corpus of segments and further improve audio browsing and retrieval. Finally, we would like to explore a temporal analysis of our data and model long term topic shifts in the hidden factors and language models.

7. REFERENCES

[1] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine Learning*, 1999.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[3] Daniel Gildea and Thomas Hofmann. Topic-based language models using em. *EuroSpeech-99*, pages 2167–2170, 1999.

[4] Marti A. Hearst. Context and structure in automated full-text information access. *University of California at Berkeley dissertation. Computer Science Division Technical Report*, 1994.

[5] Thomas Hofmann. Probabilistic latent semantic indexing. *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[6] P. van Mulbregt, I. Carp, L. Gillick, S. Lowe, and J. Yamron. Text segmentation and topic tracking on broadcast news via a hidden markov model approach. *Proceedings ICSLP-98.*

[7] Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.