TEXT SEGMENTATION AND TOPIC TRACKING ON BROADCAST NEWS VIA A HIDDEN MARKOV MODEL APPROACH

P. van Mulbregt, I. Carp, L. Gillick, S. Lowe and J. Yamron

Dragon Systems, Inc.
320 Nevada Street
Newton, MA 02460

ABSTRACT

Continuing progress in the automatic transcription of broadcast speech via speech recognition has raised the possibility of applying information retrieval techniques to the resulting (errorful) text. In this paper we describe a general methodology based on Hidden Markov Models and classical language modeling techniques for automatically inferring story boundaries (*segmentation*) and for retrieving stories relating to a specific topic (*tracking*). We will present in detail the features and performance of the Segmentation and Tracking systems submitted by Dragon Systems for the 1998 Topic Detection and Tracking evaluation.

1. INTRODUCTION

Over the last few years Dragon, like a number of other research sites, has been developing a speech recognition system capable of automatically transcribing broadcast speech. With the recent advances in this technology, a new source is becoming available for information mining, in the form of a continuous stream of errorful, unsegmented text. Applying standard information processing techniques to this data, such as topic tracking, requires that this text first be segmented into topically homogeneous blocks. Unlike newswire, typical automatically transcribed audio data contains no information (other than pauses in the audio) about how the stream should be divided up.

In [1], we introduced a new approach to text segmentation and topic tracking, one based on Hidden Markov Models (HMMs) and classical language modeling. In that paper we applied the method to segment text from the Topic Detection and Tracking (TDT) Pilot Study corpus, made up of Reuters newswire and manually transcribed CNN news stories. In [2], we discussed segmentation results on a subset of the TDT2 corpus, as well as tracking results on both manually and automatically segmented text. Here we discuss recent improvements to both the segmenter and the tracker, and present results from the 1998 TDT2 evaluation.

The experiments described here on segmentation and topic tracking were carried out on the TDT2 corpus and evaluated following the procedures set out in the TDT Evaluation Plan. Both are available through the Linguistic Data Consortium (LDC) at the University of Pennsylvania.

This paper is organized as follows. Details of the TDT2 corpus are given in Section 2. The segmenter is described in Section 3, and the tracker in Section 4. In Section 5 we describe a large number of experiments, many done as part of the 1998 TDT evaluation and many others as contrasts.

2. THE TDT2 CORPUS

The DARPA Topic Detection and Tracking program is concerned with the development of information processing technology that can applied to large streams of data, such as newswire and broadcast news. To facilitate research on the TDT tasks, the *TDT2 corpus* has been created. This corpus consists of about 60,000 news stories from newswire, television, and radio, collected by the Linguistic Data Consortium (LDC) over the period January 1998 through June 1998. A feature of this corpus, key to the tracking research, is that each story has been labeled with a binary decision as to its relevance to each of 100 topics. Tracking judgments can therefore be directly compared to human judgments.

The newswire component of the corpus was collected from the Associated Press Worldstream (APW) and the New York Times News Service (NYT). The broadcast portion includes entire shows from the Cable News Network (CNN), American Broadcasting Company (ABC), Public Radio International (PRI), and Voice of America (VOA). The broadcasts (approximately 800 hours) were transcribed both automatically and in closed-caption, the automatic version generated using a modification of Dragon's 1997 Hub-4 recognizer [3].

3. THE SEGMENTER

3.1. Overview

Our approach to the problem of segmentation is to treat a story as an instance of some underlying topic, and to model an unbroken text stream as an unlabeled

sequence of these topics. In this model, finding story boundaries is equivalent to finding topic transitions. Just as in speech recognition, this situation is subject to analysis using classic HMM techniques, in which the hidden states are topics and the observations are words or sentences.

Specifically, we build a network where each node corresponds to a unigram language model and sentence/utterance of text, and find the best path through this network. The score of each node is the score of its sentence in its language model. There is a penalty for topic-topic transitions between nodes. A search for the best hypothesis and corresponding segmentation can be done using standard HMM techniques. This segmentation does provide a label for each segment, namely the topic to which that segment was assigned in the best path, but the label may not have much meaning to a human.

3.2. Constructing the Topic Language Models

The topic language models used by the segmenter were built from the newswire and the automatically transcribed broadcasts from the January–April TDT2 data. This totaled about 15 million words spread across about 48,000 stories of average length 310 words, though the average length varied from a low of 129 for CNN, to a high of 850 for the New York Times. A global unigram model consisting of 60,000 words was built from this data.

Topic clusters were constructed by automatically clustering the stories in the training data. This clustering was done using a multi-pass k-means algorithm described in [1]. In order to prevent very common words and punctuation symbols from dominating the computation, we introduced a stop list containing 112 entries. These words did not participate in the computation of the distance measure. Removing these words from the vocabulary meant that approximately half the words in the text stream were not scored.

A topic language model was built from each cluster. We chose to model each topic using unigram statistics only. These unigram models were smoothed versions of the raw unigram models generated from the clusters. Smoothing each model consisted of performing absolute discounting followed by backoff [4] to the global unigram model; in other words, a small fixed count (about .5) was subtracted from the nonzero raw frequencies, and the liberated counts were redistributed to the rest of the words in the model in proportion to the global unigram distribution built from the training data. The raw cluster unigrams were quite sparse, typically containing occurrences of only 6,000 distinct words from the training list of 60,000 words. Words on the stop list were removed from the models.

We will frequently refer to these topic language models as *background topics* or *background models.* Note that these models could have been built from any news sources — the idea is for the clusters to represent all of English news discourse — and not be source dependent.

3.3. The Switch Penalty

If one assumes that the topic transition probabilities are independent of the topics, the transitions can be modeled with a single number, the *switch penalty,* which is imposed whenever the topic changes between segments. This is the only adjustable parameter in our system except the search beam width, which was set large enough to eliminate search errors.

The switch penalty controls the number of boundaries that are output by the system. It depends on the broadcast source and the granularity of the background models, and should be robust to slight perturbations of the models. The actual tuning was accomplished by training a system with k topics on all except a small held-out set of data, tuning the switch penalty on this held-out set, and using this switch penalty for models with k topics built on all of the allowable training data.

The switch penalty is the only parameter depending on the broadcast source, and as such the background topic models can be used on any source as long as the average document length is known.

4. THE TRACKER

4.1. Overview

In the Tracking task (a variation of the filtering task in information retrieval), a system is supplied with a few examples of stories on a particular topic of interest and is expected to automatically find subsequent examples in the stream. Specifically, a system is given as training material the first N_t examples in the evaluation corpus of stories on a particular topic (the *topic training stories*), plus all off-topic stories in the evaluation corpus prior to the last training example (*off-topic training stories*), plus all stories prior to the evaluation corpus (*background data*), and asked to return judgments on all remaining evaluation stories.

In this paper we will describe the second incarnation of a tracking system which uses standard language modeling techniques (in particular, unigram statistics) to measure document similarity. As in our earlier work [1, 2], this system is based on a simple classifier:

- Score an incoming story against a topic unigram language model built from the topic training stories
- Score the story against a discriminator language model built from the background data
- Output the difference between these scores as a relevance value, or threshold this difference to generate a decision

One of the key ways in which this system is different from its earlier incarnations is in the way we smooth the extremely sparse topic unigram models that arise from the topic training stories. We have improved the targeting procedure and introduced a variation on linear discounting that has significantly improved performance.

The nature of the TDT2 corpus makes it likely that, for any given evaluation topic, the data from which we build our multiple discriminators (the TDT2 training and development sets, or January–April 1998 data) is "contaminated" with on-topic material. For this reason we are now careful to filter such material in the construction of the discriminator models. In addition, one of these models is now targeted specifically to the tracked topic to better discriminate on-topic and close-to-topic stories.

4.2. Smoothing of the Topic Models

Our approach to the smoothing problem has focused on the use of *targeting*, in which we take a large number of language models built from the background material, find the mixture that best approximates the sparse model, and use this mixture as a smoothing distribution.

Specifically, given a sparse topic unigram model $t(w_n)$ built from the topic training data, and a set of *background models* $b^{(i)}(w_n)$, we find the best mixture

$$b(w_n) = \sum_i \lambda^{(i)} b^{(i)}(w_n) \ , \ \sum_i \lambda^{(i)} = 1 \ ,$$

such that the Kullback-Leibler distance between $t(w_n)$ and $b(w_n)$ is minimized. This leads to an implicit equation for the $\lambda^{(i)}$:

$$\lambda^{(i)} = \sum_n \frac{t(w_n)\lambda^{(i)} b^{(i)}(w_n)}{\sum_j \lambda^{(j)} b^{(j)}(w_n)} \ ,$$

which is easily solved by iteration.

In earlier versions of our system we targeted the topic unigram model against unigram models derived from clusters of stories from the background data (typically about 100 models). In this investigation, we targeted against the unigram models associated with the individual background stories (15–50,000 models — although for reasons having to do with the discriminator, some of the background stories were filtered out first; see Section 4.3). Our motivation is that a mixture based on documents can select background data more like the topic training data, and therefore generalize that data in a more realistic way compared to a mixture based on coarse clusters.

One problem that can result when targeting to individual stories is the assignment of a large proportion of the mixture probability to a small number of stories, yielding a mixture distribution which is itself sparse. To measure the sparseness of the mixture (recall it is a *probability* distribution built from a large number of components, and so may not actually contain zeros), we assign it a total count B according to

$$B = \exp\Big(\sum_i \lambda^{(i)} \log \frac{c^{(i)}}{\lambda^{(i)}}\Big) \ ,$$

where $c^{(i)}$ is the total count of background story i. (To understand this formula, consider the case in which all background stories have the same total count c. The expression for B then reduces to

$$B = c \exp\Big(-\sum_i \lambda^{(i)} \log \lambda^{(i)}\Big) \ ,$$

or c times the perplexity of the mixture weight distribution. Roughly speaking, this perplexity is the number of *stories* over which the mixture is distributed, so B represents the number of *counts* over which it is distributed.) Given a total count B, the mixture distribution is converted to counts and smoothed.

In our 1997 system we smoothed the mixture and topic unigram models by absolute discounting followed by backoff to a smoothing distribution. However, we have observed that in very sparse models, absolute discounting appears to be insufficiently aggressive at redistributing probability. For that reason we switched in the 1998 system to linear discounting (in which the amount discounted from each count is proportional to the count), with the linear discount parameter determined by requiring that the smoothed distribution have a specified internal perplexity, large enough to guarantee that the smoothed model has its counts distributed over a large number of words. Using this method, the targeted mixture model associated with each topic was smoothed with the global background distribution, and the topic model was then smoothed with the smoothed targeted mixture model.

4.3. The Discriminator

The discriminator for Dragon's previous system consisted of a large number of unigram models derived by automatically clustering the background material. For any given test story, the best scoring model from this set is the one chosen to compare to the topic model. The advantage of such a system is that an off-topic test story will tend to score well in at least one of the clusters, allowing it to be easily distinguished from the tracked topic.

What this system does not handle as well is the case of the off-topic story that shares features with the tracked topic. Consider, for example, the problem of distinguishing a story on a tobacco lawsuit brought by an individual, from a topic concerning the national tobacco settlement. Unless there is a background cluster concerned with tobacco lawsuits, the story will likely get a good tracking score.

To address this problem, the new system includes in the discriminator a model that is designed to be

"close" to the topic model without actually containing topic training data. It is expected that this model will be the best scoring of the discriminator models for on-topic and close-to-topic stories.

The obvious candidate for a "close" model is the targeted mixture model built to smooth the topic model. However, in order for the mixture model to work properly as a discriminator, it is crucial that the background from which it is derived be free of any on-topic material. Therefore, before doing the targeting described in the previous section, we build a rudimentary tracker and "track" the background stories. Any stories that score too well are presumed to be on topic, and are discarded. (One could use these high-scoring stories to supplement the topic training material, but this was not done in this investigation.) Targeting is then done only against the remaining stories.

Although we are careful to remove on-topic material from the background before targeting the mixture model, we do not remove it prior to producing the background clusters from which the other unigram models in the discriminator are derived (this would have required a clustering run in every tracking experiment, which is too costly). This means that for a given topic, one or more of the clusters may be contaminated with on-topic data. To correct for this, the set of cluster models is filtered to remove any that the targeted mixture model fails to outscore by a certain threshold on the topic training material.

4.4. Other Techniques

The tracker includes a mechanism for unsupervised adaptation on incoming stories that are highly likely to be on topic. If a story comes in that scores higher than a specified threshold, this story is added to the set of topic training stories, and the entire build procedure is rerun. This includes the preliminary tracking of the background to remove on-topic material, targeting a new mixture model to use as a smoothing distribution and as a discriminator, smoothing the topic model and the targeted mixture model, and filtering the background clusters to remove any that may be contaminated with on-topic material. Tracking then continues on the next available test story. Unsupervised adaptation had a small positive effect on performance.

5. THE 1998 TDT2 EVALUATION

5.1. Segmentation Results

The TDT2 Segmentation evaluation was conducted on the automatically transcribed (ASR) portion of the TDT2 corpus, taken from the months of May and June. This collection comprised 384 shows, 6,000 stories, and 2.2 million words. About 60% of the shows are from CNN. The ASR output has certain breaks marked; these typically correspond to silence, music, or speech with a music background, and were used to identify possible story transition times.

All results are reported using the C_{Seg} metric for measuring the quality of a segmentation. The metric takes the form

$$C_{Seg} = P_{Seg} * P_{Miss} + (1 - P_{Seg}) * P_{FalseAlarm}$$

where P_{Miss} and $P_{FalseAlarm}$ are computed with a window width of 50 words and P_{Seg} is the *a priori* probability of a segment boundary being within the window length. For the ASR portion of the TDT2 corpus, P_{Seg} is 0.3, corresponding to an average story-length of $50/0.3 \approx 165$. So $C_{Seg} = 0.3 * P_{Miss} + 0.7 * P_{FalseAlarm}$.

The Main Evaluation

Show	P_{Miss}	P_{FA}	C_{Seg}
ABC_WNT	0.3454	0.0888	0.1658
CNN_HDL	0.3094	0.1022	0.1644
PRI_TWD	0.3056	0.0670	0.1386
VOA_ENG	0.3333	0.0772	0.1540
VOA_TDY	0.3210	0.0695	0.1449
VOA_WRP	0.3448	0.0635	0.1479
Overall	0.3183	0.0835	0.1539

Table 1: Official segmentation performance on ASR data broken out by source.

There is a little variation across source. Our algorithm tends to work best on material which is mostly content, with few segues and fillers. There does seem to be a bias in the metric towards undergenerating segments, which accounts for the disparity between misses and false alarms. Typically the optimum number of generated segments that minimizes C_{Seg} is somewhere between 65% and 80% of the true number of segments.

Closed-Captioned and FDCH Transcripts

Source Condition	C_{Seg}	C_{Seg} for ABC
ASR	0.1579	0.1723
Closed Captioned	0.1138	0.1356
FDCH Transcripts		0.1515

Table 2: Overall and ABC segmentation performance according to source condition

Closed-caption text is available for all the sources, so these transcripts were tested on as a contrast. For ABC, the manual FDCH transcripts were also available, which gave a comparison between human transcriptions and automatic transcriptions.

The system performed better on the closed-caption and FDCH transcripts than on ASR text, which is not too surprising considering the opportunity for the segmenter to be misled by content-bearing recognition errors in the ASR. There appears also to be a difference

between the closed-caption and FDCH transcripts on the one source for which we have all three, with the closed-caption results better by about 10%. This might be explained by the tendency of the closed-caption transcriptions to skip segue and filler material.

The Effect of the Miscellaneous Models

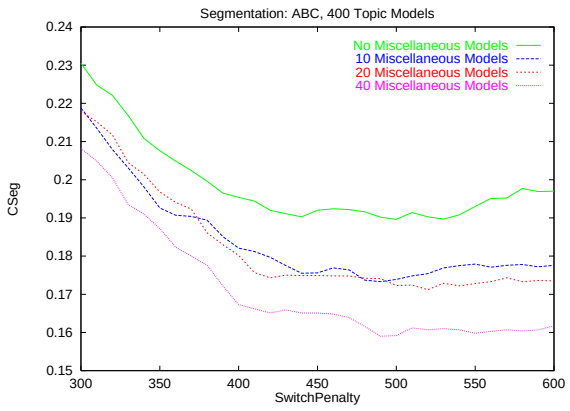

Figure 1: Varying the miscellaneous models.

The TDT ASR text contains advertisements and sports as well as news stories, and such non-news items are marked as "miscellaneous" by the LDC. Up to 10% of the data falls into this category. These stories are not scored in the TDT2 evaluation, but their presence could affect segment boundary placements close by. Hence we decided to build a number of "miscellaneous" models, and add these to the collection of topic models. The miscellaneous text was automatically clustered into 10, 20 or 40 clusters, and unigram language models were built from these clusters.

In Figure 1, the effect of the miscellaneous models is shown as the switch penalty is swept out to view a full graph. More models appear to offer an improvement.

5.2. Tracking Results

1998 System vs. 1997 System

In our development system, each topic unigram model was targeted against approximately 15,000 background stories from the TDT2 January–February data. A stop list of about 100 common words was applied before targeting. The targeted mixture model associated with each topic was smoothed with the global background distribution to an internal perplexity of 1500 (determined by tuning), and the topic model was then smoothed with the smoothed targeted mixture model, also to an internal perplexity of 1500. The discriminator consisted of the targeted mixture model and 100 automatically derived clusters of the background data. The development test material consists of the TDT2 March–April data.

Figure 2 shows a comparison of our 1998 and 1997 systems on the development test set, running under the default evaluation conditions: tracking with four story samples ($N_t = 4$) in newswire (NWT) and automatically recognized broadcast (ASR). The detection-error tradeoff (DET) plots are generated by pooling the output of the tracker from the different topic runs and sweeping a decision threshold through the story relevance scores. The fact that the 1998 plot is mostly well inside the 1997 plot indicates that the 1998 system is substantially improved over the old system.

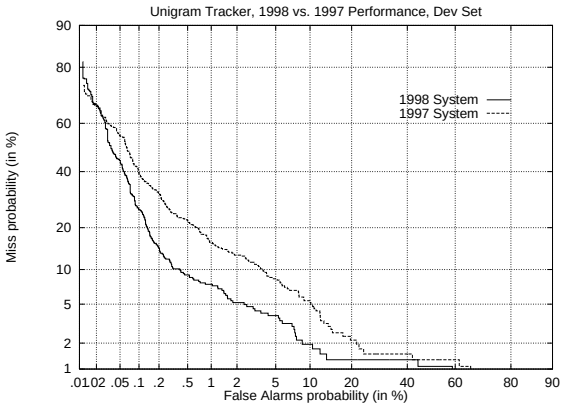

Figure 2: 1998 vs. 1997 Tracking Results

The Main Evaluation

The evaluation system is identical to the development system except that the background was taken to be the approximately 48,000 stories that comprise the TDT2 January–April data. The discriminator, once again, consisted of the targeted mixture model and 100 automatically derived clusters of the background data. The evaluation test data covers the May–June portion of the TDT2 corpus.

Figure 3 shows our performance, as a DET plot, on the evaluation data under the default test condition $N_t = 4$, along with a contrast at $N_t = 1$. (This is somewhat of an unfair comparison, as the system parameters were tuned for performance at $N_t = 4$.) Given a decision threshold on the tracking scores, the evaluation provides a single component metric, C_{Track}, for measuring system performance as a single number (smaller values are better). For the default condition, the value of C_{Track} was 0.0079.

An Interpolated System

Dragon submitted two tracking systems for evaluation, the one described here and another based on a Beta-Binomial model [5]. A third can be simply created by interpolating the output of these two. Performance was fairly insensitive to the tuning of the mixture, which was set to 50-50 based on results on the development data.

The interpolated system outperforms both of its components at all values of the decision threshold.

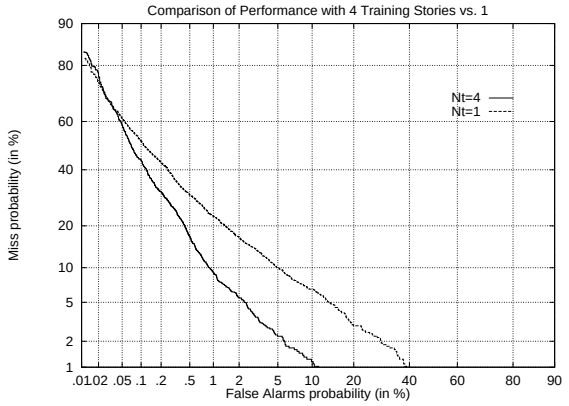

Figure 3: Effect of Reduced Training.

For the threshold chosen to minimize C_{Track}, which yielded $C_{Track} = 0.0079$ for the unigram tracker and $C_{Track} = 0.0071$ for the Beta-Binomial tracker, the interpolated tracker achieved the value $C_{Track} = 0.0062$. (All thresholds were tuned on the development data.)

5.3. Tracking on Automatically Segmented Data

One goal of the 1998 evaluation was to see what effect certain kinds of errors in the input have on performance. Figure 4 presents two comparisons: first, our official evaluation performance on newswire and automatically recognized broadcast compared to automatically recognized broadcast only, and second, our performance on automatically recognized broadcast compared to the same data with story boundaries determined automatically by Dragon's HMM segmenter. (For all runs, unsupervised adaptation was disabled in order to highlight differences due to variations in the input.)

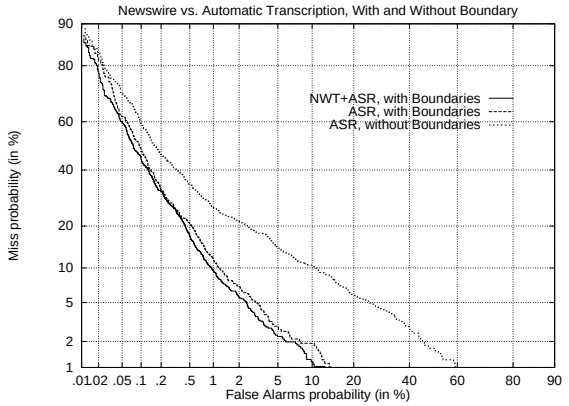

Figure 4: Newswire vs. Automatic Transcription vs. Automatic Transcription with Automatic Story Boundary Determination

Figure 4 shows that there is almost no degradation in performance associated with the automatically recognized material, despite a recognition word error rate on the order of 30%, which is consistent with other investigations. On the other hand, there is a noticeable loss of performance when story boundaries are determined by machine, particularly at low miss rates.

One contributor to this loss of performance is the fact that the minimization of C_{Seg}, as it is defined for the segmentation task, leads to segmentations with fewer than the correct number of boundaries. In order to investigate the dependence of the tracking task on the quality of the segmentation, a range of segmentations was produced, and then used as input to a simplified version of the evaluation tracker. The ratio of (number of hypothesized bundaries) to (number of correct boundaries) varied from 0.6 to 2.0.

The best tracking performance, as measured by C_{Track}, appears to occur at a segmentation ratio of 1.0 or higher. (In fact, tracking performance is quite robust to the quality of the segmentation, as long as there are more hypothesized boundaries than actual boundaries.) However, the best segmentation performance, as measured by C_{Seg}, occurs at a ratio between 0.6 and 0.8. Thus minimizing the segmentation metric C_{Seg} leads to segmentations which are not good for the follow-on task of tracking.

6. REFERENCES

[1] J.P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt, "A Hidden Markov Model Approach to Text Segmentation and Event Tracking," *Proceedings ICASSP-98, Seattle, May 1998*

[2] P. van Mulbregt, J.P. Yamron, I. Carp, L. Gillick, and S. Lowe, "Text Segmentation and Topic Tracking on Broadcast News via a Hidden Markov model approach." *Proceedings ICSLP-98, Sydney, December 1998*

[3] L. Gillick, Yoshiko Ito, Linda Manganaro, Michael Newman, Francesco Scattone, S. Wegmann, Jon Yamron, Puming Zhan, " Dragon Systems' Automatic Transcription of New TDT Corpus," *Proceedings of Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, Feb 1998.

[4] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," in *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400–401, March, 1987.

[5] S. Lowe, "The Beta-Binomial Mixture Model and Its Application to TDT Tracking and Detection," *Proceedings of the DARPA 1999 Broadcast News Workshop*, February 1999.