Dragon's Tracking and Detection Systems for the TDT2000 Evaluation

J.P. Yamron, S. Knecht, P. van Mulbregt

Dragon Systems
320 Nevada Street
Newton, MA 02460

ABSTRACT

We describe the tracking and detection systems submitted by Dragon for the TDT2000 evaluation. Our research focus is on improving the distance measure between story and story collection, a computation which is central to many of the TDT tasks. In our tracking engine, we improved the measure by strengthening our targeting procedure, and introduced unsupervised adaptation on high-scoring test stories. Our detection engine uses a new measure, developed under tracking experiments, that performs 15–20% better than the previous measure, and makes use of targeting in a computationally tractable way. In both tasks, we explore the effect of automatically segmenting broadcast news stories and of introducing word stemming.

1. INTRODUCTION

The focus of our TDT research [1, 2, 3, 4, 5, 6, 7, 8, 9] is the problem of modeling topical content, and our approach is to design, for a collection of stories defining a topic, a statistical model of story generation within that topic. Such a model can be used to evaluate the probability that a test story is on the given topic, in essence providing a measure of story-topic distance. We believe this distance measure is a central component of systems designed for TDT, and as such, improvements in the design of the measure can be leveraged to yield performance gains in a variety of TDT tasks.

For the TDT2000 evaluation, Dragon submitted systems for Tracking and Detection. For Tracking, we explored the effect of a number of techniques and conditions on performance, including changes to targeting, unsupervised adaptation, using automatically generated story boundaries, and word stemming. For Detection, we incorporated a new story-cluster distance measure inspired by tracking improvements in the TDT1999 evaluation, explored a technique for indirectly smoothing the clusters by instead smoothing the test stories via targeting, and tested the effects of using automatically generated story boundaries and word stemming.

In all experiments described here, training data was taken from the first half (January–March) of the TDT2 corpus, English only. Our development test set was the second half (April–June) of the TDT2 corpus, English only. The test set contained 70 topics with at least one annotated story, all of which were used in our tracking experiments, and all but one (eliminated due to multiple annotations) of which were used in our detection experiments.

2. TRACKING

Dragon's tracking system [1, 4, 6, 7] consists of two statistical models: a *topic model* built from the supplied topic training stories, and a *discriminator model* built from available background material. The *tracking score* of a test story is defined as the log ratio of its score in these two models—in other words, the log ratio of the probability that the the topic model generated the story *vs.* the probability that the story was generated by the discriminator. This tracking score can be interpreted as the distance measure between a story collection (the topic training stories) and a test story.

In previous evaluations, our Tracker was a combination of two systems, one based on unigram statistics and another based on the beta-binomial distribution [7, 8]. An analysis of the behavior of these two systems shows that the Unigram Tracker tended toward better performance at low decision threshold, while the Beta-Binomial Tracker was best at high threshold. Since the TDT evaluation is geared toward performance at lower thresholds, we chose to concentrate our efforts on the Unigram Tracker for this evaluation.

2.1. Unigram Tracker

In this system a topic is modeled as a simple unigram distribution, $p(w)$. The probability that a test story T was generated by $p(w)$ is just the product of the probabilities of each word being drawn from this distribution,

$$p_T = \prod_{w \in T} p(w) \ .$$

In our implementation, we incorporate a stop list of common words.

The topic model in the Unigram Tracker consists of a unigram distribution built from the topic training stories. A key problem is that this model is based on extremely sparse statistics (four training documents, for example, typically contain fewer than 1000 words total), hence smoothing is required. Our smoothing algorithm for the very sparse topic training data has, for some time now, consisted of two steps:

- A *targeting* step [6], in which a unigram mixture of background stories is constructed that resembles the unigram model generated from the topic stories. In the experiments described below, we used the 26683 stories comprising the training data as the background material.
- A smoothing step, in which the targeting mixture is incorporated into the topic model.

The discriminator model is a unigram distribution built from a large amount of backgrounds material. We used a flat mixture of the counts of the 26683 training stories, after first running a simple tracker over this material to remove stories that might be considered on-topic.

2.2. Experiments

The tracking experiments we describe below, unless specified otherwise, were performed under the following conditions:

- One training story
- Newswire and automatically transcribed broadcast news
- Manually labeled boundaries for the test stories (although a comparison with automatically labeled boundaries is shown below)
- English only

These differ from the TDT2000 required conditions, which specify manually transcribed broadcast news and English plus Mandarin test stories.

Results are presented as Detection Error Tradeoff (DET) plots of misses (stories falsely labeled as off-topic) *vs.* false-alarms (stories falsely labeled as on-topic).

Automatically labeled story boundaries. We expected some loss of performance when we replaced the manually labeled story boundaries used in our tests by the automatically labeled boundaries supplied by IBM; surprisingly, the performance in these two cases was virtually indistinguishable. This is partially a tribute to the quality of the segmentation, but also reflects the fact that the boundaries were somewhat overgenerated; previous work by Dragon [5] has shown that tracking performance on automatically segmented stories is optimized when the segmenter overgenerates.

Stemming. One technique to improve word statistics that is often used in IR is *stemming,* or replacing different forms of a word in training and test sets with a single form. Dragon's TDT systems have not included the use of stemming, for no reason other than the fact that it is not a common procedure in speech recognition.

Figure 1 shows the effect of stemming on the performance of the Tracker on the development test data. The difference in performance is small except for a noticeable gain (mainly coming from the smaller topics) at low decision thresholds. This is promising, but in other experiments in which the topic model was trained from four stories instead of one, this gain becomes a small loss. We decided not to use stemming in the TDT2000 evaluation until we better understood the effect; however, some of the potential gain was probably achieved by other means below.

Unsupervised adaptation. It seems evident that accurately detecting at least some on-topic stories in the incoming stream should be possible with the current tracking technology, and that adapting the topic model on these stories should improve tracking performance. Nevertheless, previous attempts by Dragon to do just this have not been successful [9]. Typically, adaptation has improved the performance on some larger topics at the expense of smaller topics.

One hypothesis is that models for smaller topics benefit from being narrowly defined by a small amount of data—the few on-topic stories in the incoming stream are well-modeled by the few (or even one) topic training story. Large topics, on the other hand, are large probably because they encompass a broad set of sub-topics, and therefore benefit from well-guided generalization. In order for unsupervised adaptation on incoming stories to work, then, a tracking

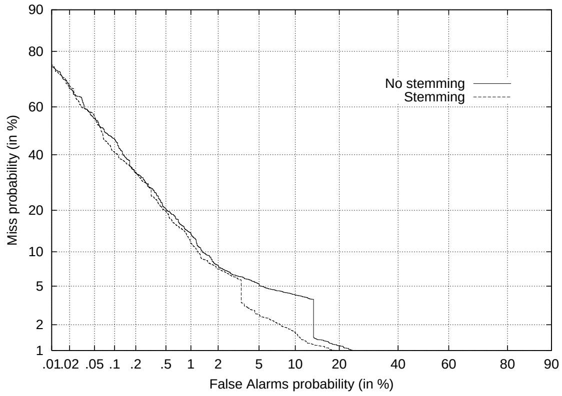

Figure 1: Effect of Stemming on Tracking Performance

system needs to be able to determine whether a topic is small or large.

For the TDT2000 evaluation, we took the following approach: first, the system only adapts on stories scoring above a fairly high threshold, to reduce the likelihood of adaptation on an off-topic story; second, adaptation is deferred until a minimum number of incoming stories meeting the high threshold have been accumulated; third, there is a maximum number of stories on which a topic may adapt. (The threshold, the deferral number, and the maximum are tuned.) The idea is that large topics will accumulate the required number of high-scoring stories and adapt, while small topics will fail to accumulate enough stories and remain narrowly defined.

Figure 2 shows the effect of incorporating unsupervised adaptation, when the deferral number and adaptation maximum are set to 5. The plot splits out the behavior of large topics (the 16 topics in the development set with 25 or more on-topic stories in the test set) from small topics (the 28 topics with between 1 and 24 on-topic stories). (The are an additional 26 topics with no on-topic test stories.) There is an improvement in performance on large topics at all decision thresholds, and only a small degradation in performance on smaller topics.

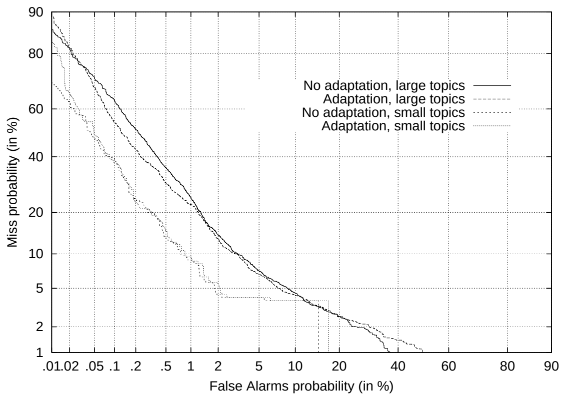

Figure 2: Effect of Unsupervised Adaptation

Stronger targeting. In the TDT1999 evaluation, the targeting mixture was not allowed to contain stories very close in topic to the topic training material. This was not desirable, but was forced on us due to an overlap between development training and test material in the TDT2 corpus that caused targeting to fail unless these stories were stripped out. For the TDT2000 evaluation this problem was rectified, and these topically close stories are now allowed in the mixture, and therefore in the smoothing of the topic model. As one might expect, including such material leads to a better trained topic model with better performance.

Figure 3 shows the sharp gain at low decision threshold when the targeting contains the topically close stories. The improvement at low threshold indicates that the topic model is better able to generalize, and can therefore recognize topically related material that shares few features with the original topic training story.

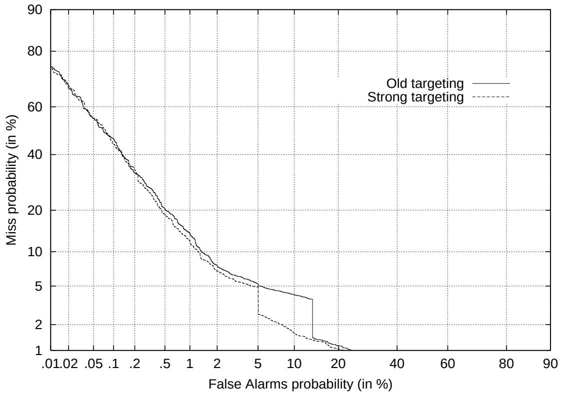

Figure 3: Using More Topical Material in Targeting

2.3. System for the TDT2000 Evaluation

The TDT2000 tracking system incorporated the stronger targeting and does unsupervised adaptation on high-scoring incoming stories. Figure 4 shows the performance of this system compared to the performance of the 1999 system. The sharp gain at low decision threshold comes mainly from the smaller topics and is the result of using stronger targeting; the smaller, persistent gain at the middle decision thresholds comes mainly from the larger topics and is the result of adaptation, the effect of which seems to be enhanced when combined with the stronger targeting.

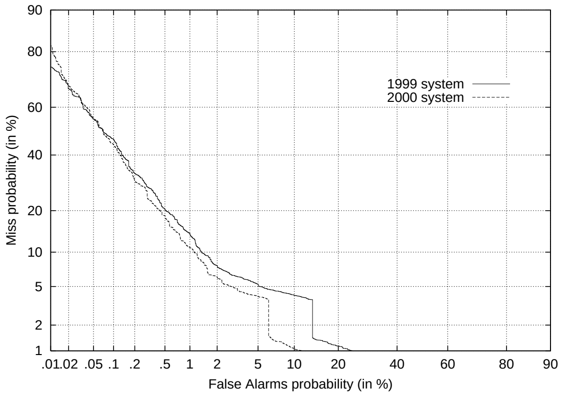

Figure 4: Performance of 1999 System *vs.* 2000 System

3. DETECTION

Our detection system [2, 7] is based on a simple k-means clustering engine, that operates as follows:

- At any given point there are k story clusters, each cluster characterized by a set of statistics. For the next available story, determine its distance to the closest cluster, and if this distance is below a threshold, insert the story into the cluster and update the cluster statistics. If this distance is above the threshold, create a new cluster.
- Loop through the stories again, but now consider switching each story from its present topic to each of the others, based on the same distance measure as before. Some clusters may vanish; additional clusters may need to be created. Repeat this step as often as desired.

Note that this algorithm relies on the existence of a cluster–story distance measure. An obvious candidate for such a measure is the tracking score described previously—in other words, at each point in the clustering process that a story is compared to a cluster, the cluster can be treated as a story collection for which we can build a "topic model," and, given a discriminator model built from background data, the tracking score can be used as the cluster–story distance.

One obstacle we have run into is that the computational complexity of the clustering task makes it difficult to take advantage of our best tracking measures in detection. In the clustering task, the clusters are constantly being updated, with each update triggering (in the language of tracking) a "topic model build." These topic builds must be kept simple to keep the Detection task tractable.

3.1. Measures

The cluster–story measure used for detection can be described as follows:

- Each cluster is treated as a "topic" (in the tracking sense), and a topic model is derived from the unigram counts making up the cluster by simple smoothing.
- A discriminator unigram model is build from background material.
- The cluster–story measure (like the tracking measure) is just the log ratio of the probability that the the cluster model generated the story *vs.* the probability that the story was generated by the discriminator.

For the TDT1999 evaluation, the cluster unigram distribution was smoothed via absolute discounting [10] with a general unigram distribution built from background material. The resulting cluster–story measure, when evaluated as a tracking measure, is terrible; a high false-alarm tail indicates a severe inability to generalize beyond the stories making up the cluster.

Obvious candidates to replace this detection measure are the measures used successfully in tracking. Unfortunately, the best tracking measures are too expensive, involving in one case a targeting against thousands of background documents, and in the other the computation of beta-binomial statistics. Nevertheless, because the tracking task is computationally simpler, it is a good testbed for cluster–story measures, allowing us to try out more variations.

Work leading up to the TDT1999 Tracking evaluation yielded one candidate measure that performed significantly better on the tracking task than the 1999 detection measure, and which was also simple to calculate. Like the 1999 detection measure, the topic model consists of a simple smoothing of the topic unigrams with a background distribution, but the smoothing is done via interpolation instead of backoff. This is the measure used in the TDT2000 Detection evaluation. Figure 5 shows the performance of this measure on a tracking task, compared to the 1999 detection measure (which is substantially worse) and the 2000 tracking measure (which is better, but too computationally expensive for detection).

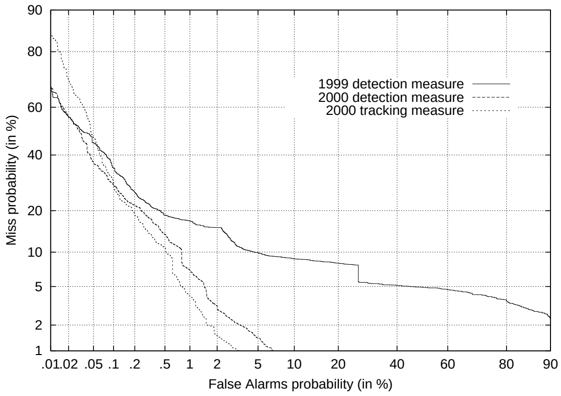

Figure 5: Detection Measure *vs.* Tracking Measure

3.2. Experiments

The detection experiments we describe below, unless specified otherwise, were performed under the following conditions:

- Deferral of 10 files
- Newswire and automatically transcribed broadcast news
- Manually labeled boundaries for the test stories (although a comparison with automatically labeled boundaries is shown below)
- English only

These differ from the TDT2000 required conditions, which specify automatically labeled boundaries and English plus Mandarin test stories.

For each experiment, the clustering engine is run multiple times with varying system parameters to produce runs with different numbers of clusters. Results are presented as plots of the detection cost function C_{det} *vs.* number of clusters produced.

The new measure. Figure 6 shows a comparison of the detection performance of the old *vs.* the new measure. Despite the extremely jumpy behavior, the new smoothing is clearly superior, and achieves gains of 15–20% in the value of C_{det}.

The wide variability in the value of C_{det} as the number of clusters is varied is a problem for both measures, and makes it difficult to choose an operating point (*i.e.,* tune the number of output clusters) that optimizes performance. Some of the modifications described below have the effect of smoothing this behavior somewhat.

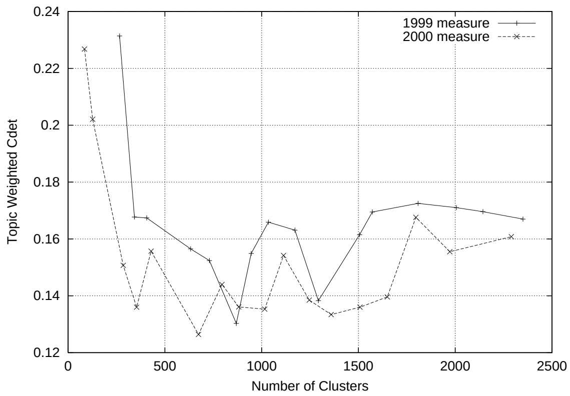

Figure 6: Performance of 1999 Measure *vs.* 2000 Measure

Automatically labeled story boundaries. Unlike tracking, in which the introduction of automatic story segmentation had virtually no effect on performance, detection performance degraded significantly when story boundaries were automatically determined. Figure 7 shows the effect of introducing automatically labeled boundaries on the new measure, and reveals a typical loss of about 20% in the value of C_{det}.

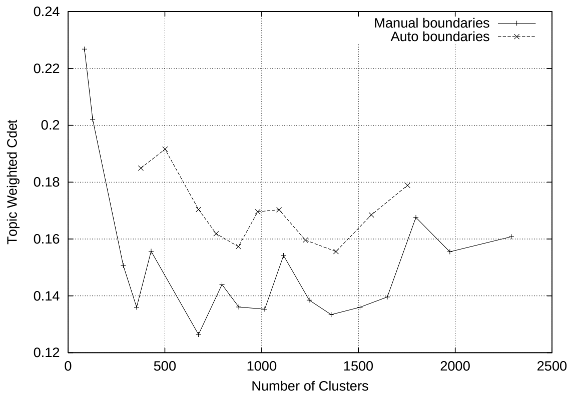

Figure 7: Manual *vs.* Automatic Story Boundaries

Stemming. As for tracking, the result of applying stemming in our detection system is a mixed bag. The behavior is shown in Figure 8. When manually labeled story boundaries are used, stemming provides a small but clear win in the region of minimum C_{det}; perhaps more importantly, it appears to result in smoother performance as the number of output clusters varies. When stories are automatically segmented, however, performance suffers. Since the required evaluation condition is with automatically labeled boundaries, we chose not to use stemming in the evaluation.

Use of targeting. We made an attempt to take advantage of targeting to improve the smoothing used in our detection system. The computation is too expensive to incorporate into the update of the clusters, so instead we used targeting to smooth the incoming stories. This is backwards—we *should* be broadening the cluster distributions to

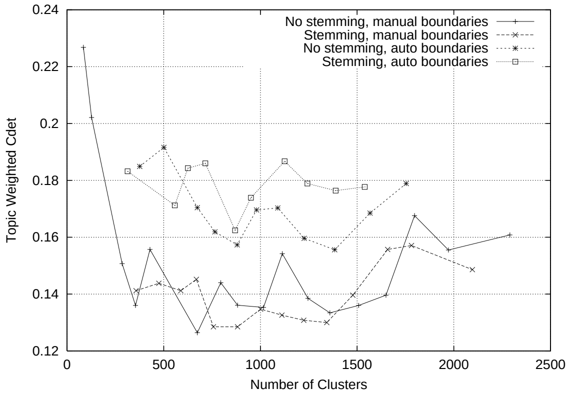

Figure 8: Effect of Stemming on Detection Performance

better score the sparse incoming stories, not smoothing the incoming stories to create more overlap with the clusters. Nevertheless, smoothed incoming stories do lead to smoother clusters, since the story statistics are merged into the cluster to which they are assigned (although the result is not the same as if the cluster itself had been smoothed by targeting).

For each story in the development test corpus we computed a targeting mixture distribution. This distribution was then used to smooth via interpolation the story from which it was derived. Detection was run using the smoothed stories as the incoming stream instead of the original stories. The interpolation weight was globally tuned, and the best choice (15% weight on the targeting mixture) is shown in Figure 9, compared to the run without targeting. The result is slightly worse, but substantially flatter and more predictable. On the basis of this it was submitted as our primary system, with the untargeted run submitted as a secondary.

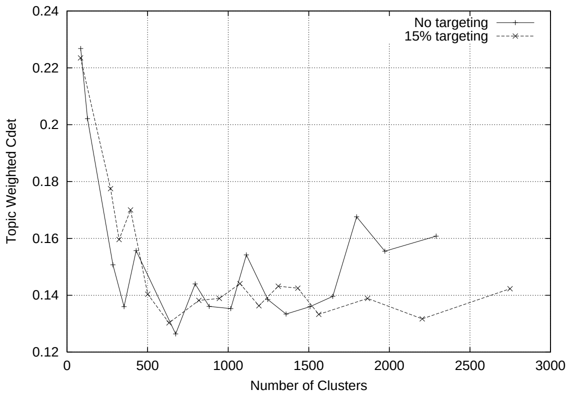

Figure 9: Targeting the Incoming Stories

References

1. J.P. Yamron, I. Carp, L. Gillick, S.A. Lowe, and P. van Mulbregt, "Event Tracking and Text Segmentation via Hidden Markov Models," *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop,* Santa Barbara, December 1997.
2. J. Allan, J. Carbonell, G. Doddington, J.P. Yamron, and Y. Yang, "Topic Detection and Tracking Pilot Study: Final Report," *Proceedings of Broadcast News Transcription and Understanding Workshop,* Lansdowne, Virginia, February 1998.
3. J.P. Yamron, I. Carp, L. Gillick, S.A. Lowe, and P. van Mulbregt, "A Hidden Markov Model Approach to Text Segmentation and Event Tracking," *Proceedings ICASSP-98,* Seattle, May 1998.
4. P. van Mulbregt, J.P. Yamron, I. Carp, L. Gillick, and S.A. Lowe, "Text Segmentation and Topic Tracking on Broadcast News via a Hidden Markov Model Approach." *Proceedings ICSLP-98,* Sydney, December 1998.
5. P. van Mulbregt, I. Carp, L. Gillick, S.A. Lowe, and J.P. Yamron, "Segmentation of Automatically Transcribed Broadcast News Text," *Proceedings of the DARPA Broadcast News Workshop,* February 1999.
6. J.P. Yamron, I. Carp, L. Gillick, S.A. Lowe, and P. van Mulbregt, "Topic Tracking in a News Stream", *Proceedings of the DARPA Broadcast News Workshop,* February 1999.
7. S.A. Lowe, "The Beta-Binomial Mixture Model and Its Application to TDT Tracking and Detection," *Proceedings of the DARPA Broadcast News Workshop,* February 1999.
8. S.A. Lowe, "The Beta-Binomial Mixture Model for Word Frequencies in Documents with Applications to Information Retrieval," *Proceedings of Eurospeech '99,* Budapest, September 1999.
9. J.P. Yamron, L. Gillick, S. Knecht, S.A. Lowe, and P. van Mulbregt, "Statistical Models for Tracking and Detection," *Proceedings of the DARPA TDT Workshop,* February 2000.
10. H. Ney, U. Essen, and R. Kneser, "On Structuring Probabilistic Dependences in Stochastic Language Modelling," *Computer Speech and Language,* 8:1–38, 1994.