IR-233

Detection As Multi-topic Tracking

James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst USA

Abstract

The topic tracking task from TDT is a variant of information filtering tasks that focuses on event-based topics in streams of broadcast news. In this study, we compare tracking to another TDT task, detection, which has the goal of partitioning *all* arriving news into topics, regardless of whether the topics are of interest to anyone, and even when a new topic appears that had not been previous anticipated. There are clear relationships between the two tasks (under some assumptions, a "perfect" tracking system could "solve" the detection problem), but they are evaluated quite differently. We describe the two tasks and discuss their similarities. We show how viewing detection as a form of multi-topic parallel tracking can illuminate the performance tradeoffs of detection over tracking.

1 Introduction

Topic Detection and Tracking (TDT) is a body of research and an evaluation paradigm that addresses event-based organization of broadcast news. The TDT evaluation tasks of tracking, cluster detection, and first story detection are information filtering technology in the sense that they require that "yes or no" decisions be made on a stream of news stories before additional stories have arrived. Each task has an established and distinct evaluation methodology to help drive research on the underlying technology.

We are motivated in this work by the belief that aspects of one problem are often illuminated by recognizing its variants in other problems. As an example, Allan and Lavrenko [Allan et al., 2000c] showed that the relationship between two of those TDT tasks (tracking and first story detection) was so strong that with certain assumptions, effectiveness on one task could be predicted from performance on the other. That result strongly suggested that satisfactory solutions to the first story detection task were not likely to be achieved using current tracking-like approaches—i.e., that the research community needed to look elsewhere to realize substantial improvements in effectiveness for one task.

In this study, we explore the relationships and differences between tracking and cluster detection (called "detection" for the remainder of this study), and in how they are evaluated. We show how the detection task is a special form of multi-topic tracking, an unsupervised system that "tracks" an arbitrarily large number of topics simultaneously. We will evaluate a detection system as if it were a tracking system, and compare the result to a "true" tracking system.

There is a general feeling in the TDT research community that solutions to one of the three filtering tasks are likely to carry over to the other tasks. The earlier work mentioned above showed that although

tracking and first story detection are related, that relationship can be a dangerous trap since it is a dead-end: tracking technology is probably not sufficient to address first story detection effectively. In this work, we will argue that tracking and detection are not as similar as they might appear at first, that solutions to one task may help with another, but will not necessarily solve the other.

In Section 2 we provide details about the TDT tasks that we are investigating, describing tracking, detection, and how detection is related to tracking. We continue in Section 3 by talking about evaluation measures that are used in TDT and filtering and how they are related. In Section 4 we describe the specific experimental setup that we used to evaluate detection and tracking. We then discuss in Section 5 how the evaluations and the systems compare. We discuss our conclusions and propose future directions for research in Section 6.

2 Tasks

TDT investigation has been carried out over five years by about a dozen academic and industrial research institutions, and explored in the context of four annual "cooperatively competitive" evaluations sponsored by the U.S. government [Allan et al., 2001, Allan et al., 1998a, DARPA, 1999, NIST, 2000a, NIST, 2000b], with several others already being planned. All problems within TDT envision a stream of constantly arriving news stories, coming from a wide range of sources, as text or audio, and in multiple languages. The goal of TDT is to organize that stream of media in three primary ways:

- Audio sources need to be segmented into individual stories. This task, called *segmentation* is not needed for newswire sources since their data is already divided into coherent stories. Segmentation is not the focus of this study, and is not discussed further. We assume that all arriving stories are correctly segmented (the corpus used in the experiments provides human-generated story boundaries).

- Stories on the same topic need to be grouped together. This task is called *detection*. It is totally unsupervised—the system must group stories into single-topic clusters without any human feedback. When new topics appear in the stream of news, the system must automatically decide to create a new cluster. A variation on this task, *first story detection*, focuses entirely on the ability to recognize when a new cluster must be started—i.e., when the first story of a news topic appears. In this study, we are concerned only with the clustering version of the task.

- If a small number of stories are designated as being on the same topic, the system must *track* that topic to find all following stories. Typically, tracking will be given up to four stories that are identified as "on topic." The major distinction between tracking and detection is that some number of stories are *known* to be on-topic, and that most of the news stories can be discarded as totally irrelevant—in detection, every story must be clustered.

The following sections describe the tracking and detection tasks in more detail. Then we discuss how detection relates to tracking.

2.1 Topic tracking

The TDT tracking task is fundamentally similar to TREC's filtering task [Voorhees and Harman, 2000]. Each begins with a representation of a topic and then monitors a stream of arriving stories,[1] making decisions about stories as they arrive. Stories are assigned a score for that topic and, if the score is high enough, are tracked or retrieved—i.e., the score threshold determines a "yes or no" decision for each story. The specifics of the tasks are slightly different:

- The topic in filtering is a subject-based query. It is represented by an explicit query, though sometimes is augmented with sample relevant (and non-relevant) stories.
- The topic in tracking is an event-based news topic. It is never represented by an explicit query, but only by a small number of training stories (e.g., $N_t = 4$) that are known to be on the same topic.
- There is no user feedback after tracking begins. Systems may adapt based on their "guesses" that a story is on topic, but they do not get human confirmation that they were correct.

A tracking system, then, is provided with a small number, N_t, of on-topic training stories. N_t has ranged from one to eight in evaluations, with four being the most commonly used value in research. The system's task is to analyze those stories and *automatically* identify the news topic that is being discussed, taking into account confounding text such as reporter banter and short or lengthy references to other news topics. No human intervention is permitted to clarify the topic or its bounds.

The system is then provided with a stream of news stories, grouped into roughly half-hour news shows, and must decide for every story whether it not it is on the same topic as those N_t training stories. The decision consists of a confidence score as well as a hard "yes or no" decision for each story. All stories in that group must be decided upon before the next set is provided to the system.[2]

A system is expected to track multiple topics, although the evaluation paradigm requires that tracking be done independently. That is, a tracking system should not use knowledge about tracking progress within one topic to make decisions on another. (Because tracking is usually done with $N_t > 1$, there is human-provided information implicitly available in other topics. The independent tracking restriction prevents a system from leveraging that "illegal" information.)

2.2 Topic detection

The TDT (cluster) detection task, on the other hand, effectively processes all news topics simultaneously. The goal of this task is to partition all arriving news into "bins" depending on the news topic being discussed: all stories about a particular earthquake should be grouped together, all stories about other earthquakes or about entirely different topics should appear in differing bins. An important component of the detection task is the recognition of when no bins are suitable so that a new bin must be created—i.e., the recognition that a new topic has appeared in the news.

[1] In information filtering, the objects of interest are usually referred to as "documents," while in TDT they are always called "stories" because of the restriction to news. This study focuses on TDT (albeit in the context of filtering), so we will use TDT terminology even though it is somewhat more restrictive.

[2] Variations of the task allow a longer deferral—e.g., allowing the decisions for a set to be postponed until 10 additional sets have arrived. We are not considering those variations here.

Unlike tracking, the detection task allows *no* supervision whatsoever. The only helpful information about stories and topics comes from pre-annotated training data that is likely to share only a few topics with the news being processed. Humans are not permitted to correct or confirm a system's output while it is running.

A major distinction between tracking and detection is an underlying assumption about whether or not stories can be on multiple topics. Tracking, because topics are independently processed, allows the possibility that a story could be tracked by multiple topics—i.e., that the story could discuss more than one topic. Detection, on the other hand, forces a strict partitioning of the stream of news stories, requiring that each story be assigned to a *single* cluster (topic).

2.3 Detection as tracking

An obvious relationship between detection and tracking appears when $N_t = 1$ for the latter task. In that case, a tracking system is expected to use that single story to find all remaining stories on the same topic. Note that a *perfect* tracking system could then almost solve the detection task. Specifically, the system would start with the first story, assign it a cluster, and then start tracking that single story. The first time a story did not fit the cluster being tracked, the system would create a second cluster and start tracking both clusters simultaneously. And so on: each time a new story did not track, it would begin a new cluster and start a new tracking process. The only aspect of detection that is not addressed is how to handle stories that are tracked by multiple topics.

In that way, detection can be viewed as a form of parallel tracking of multiple topics. Rather than tracking a subset of the news topics, it is required to "track" every topic that appears in the news. If we relax the partitioning requirement and allow a story to fall into multiple clusters, the parallel between tracking and detection is even clearer.

We will use this connection between tracking and detection below to convert one problem to the other in evaluation. That is, we will run a standard detection system and then evaluation it as a tracking system. We will show advantages to this approach over the typical TDT evaluation of detection.

3 Evaluation measures and approaches

Filtering-style tasks can and have been evaluated using a wide range of measures. All measures are set-based measures and provide some notion of effectiveness when applied to an output set. Most of the measures can also be calculated at a range of threshold values to show how they trade off against one another.

3.1 Set-based measures

Most filtering and tracking measures can be defined in terms of the well-known contingency table:

	Retrieved	Not retrieved
On-topic	A	B
Off-topic	C	D

A, for example, represents the number of stories that were both retrieved *and* on-topic, $A + C$ is the total number of stories retrieved, $A + B$ is the size of the on-topic set, and so on. The following commonly used measures from IR and TDT can be expressed in terms of those numbers:

Recall	$\frac{A}{A+B}$	Proportion of on-topic material that is retrieved
Precision	$\frac{A}{A+C}$	Proportion of retrieved material that is on-topic
Miss	$\frac{B}{A+B}$	Proportion of on-topic material that is not retrieved; this value is the same as 1–Recall
False alarm	$\frac{C}{C+D}$	Proportion of off-topic material that is retrieved; this value is also called fallout
Richness	$\frac{A+B}{A+B+C+D}$	Proportion of the collection that is on-topic; this measure is also called generality[Salton and McGill, 1983]
T9P	$\frac{A}{\max(\text{target}, A+B)}$	Precision modified for TREC-9 filtering to limit results to a minimum possible value by setting the target to, e.g., 50[Robertson and Hull, 2001]. The main reason for the minimum is to support cross-topic averaging.
T9U	$\max(2A - D, \text{minu})$	Utility score designed for TREC-9 filtering that cannot fall below a particular value. The main reason for the minimum is to support cross-topic averaging.

The TDT cost measure is a form of utility measure that combines the likelihood of errors (miss and false alarm) with prior probabilities and costs of errors:

$$\text{Cost} = C_{\text{miss}} \cdot P(\text{miss}) \cdot P(\text{target}) + C_{\text{fa}} \cdot P(\text{fa}) \cdot P(\text{non-target})$$

where $P(\text{miss})$ and $P(\text{fa})$ are the conditional probabilities of a miss and false alarm (i.e., $\frac{B}{A+B}$ and $\frac{C}{C+D}$ above), $P(\text{target})$ and $P(\text{non-target})$ are *a priori* probabilities of a story being on- or off-topic (calculated from training data), and C_{miss} and C_{fa} are the costs of a miss and false alarm, respectively. The costs can be adjusted to reflect different "real world" assumptions about user tolerances. For TDT 2000, those numbers were set to:

Measure	Detection	Tracking
C_{miss}	1.0	1.0
C_{fa}	0.1	0.1
$P(\text{target})$	0.02	0.02

Note that given those numbers as constants, this yields the cost measure:

$$\text{Cost}_{\text{TDT2000}} = 0.02 \cdot \frac{B}{A+B} + 0.002 \cdot \frac{C}{C+D}$$

This measure is similar in spirit to T9U, but is expressed as cost rather than utility, is based on error rates rather than actual counts of errors, and incorporates the prior probability of an error happening.

Given the TDT cost measures, a system could in theory always choose "no," yielding a 100% miss rate, and a cost of 0.02. Or, even better, it could always choose "yes" and achieve a cost of 0.002. So that numbers can be compared to "no effort" approaches, TDT cost measures are normalized by the minimum of those two values. So the normalized cost of always saying "yes" would be 1.0, and of always saying "no" would be 10.0. A "useful" approach must have a normalized cost below one.

Numerous other measures have also been proposed, including measures such as normalized recall and precision, F, E, the number of topics that retrieved no stories, etc., all of which combine the measures

above [van Rijsbergen, 1979, Salton and McGill, 1983, Robertson and Hull, 2001]. We do not consider those measures in this study, even though some of them have appeared in TDT and filtering research reports [Allan et al., 1998b, Yang et al., 1998, Robertson and Hull, 2001].

3.2 Tradeoff measures

One problem with the set-based measures above is that they require careful selection of a cutoff mechanism for deciding which stories to include and which to omit. Most modern systems generate confidence scores that reflect the likelihood that a story is on-topic, and then determine a treshold on that score for making the hard "yes or no" decision.

The well known recall/precision graph portrays the quality of that threshold by showing how the two measures trade off against each other as the threshold varies. A high threshold means that few off-topic stories are included in the set (precision is good); a low threshold means that few on-topic stories are omitted (recall is good). Any threshold value corresponds to a specific point on that graph. Using this graph, researchers can focus on improving error tradeoff independently of isolating the threshold. Recall/precision graphs are generally created [van Rijsbergen, 1979, Salton and McGill, 1983] by calculating the graphs for each topic individually, interpolating them to common recall points, and then averaging the graphs. Each topic is given equal weight, and because the averaging is done at common recall points, it is not important what specific score yielded that rate for each topic. That is, it is not necessary that the scores be consistent across topics.

A Detection Error Tradeoff (DET) plot is similar to a recall/precision graph, except that the axes represent miss and false alarm error rates [Martin et al., 1997]. That means that a perfect system will have a DET curve in the lower left of the graph, and as the curve moves toward the upper right, effectiveness is dropping. As with recall and precision, the two error measures have been shown empirically to vary inversely. To calculate the curve, each each *score* value, the miss and false alarm error rates are calculated for each of the topics and then averaged. Each topic is weighted equally at each point on the DET curve. However, because the averaging is done where the scores are equal, it is critical that the scores be comparable across topics.[3]

The axes of the DET curve are on a Gaussian scale—i.e., such that the normal deviate is linear (every standard deviation from the mean advances the same distance on the axes). The result of this is that if the distributions of on-topic and of off-topic story scores are normal, then the resulting DET curve will be a straight line.

Figures 1 and 2 show examples of a DET curve and a recall/precision graph, respectively. The DET curve is a variation on operating characteristic curves [Swets, 1988]. The DET curve was adopted for TDT, but the ideas behind it are far from new in the IR community. The derivation of Swets' model of evaluation [van Rijsbergen, 1979] used the same approach, for example.

[3]There is no particular reason that a DET curve could not be calculated by averaging at common false alarm (or miss) rates, or that a recall/precision graph could not be calculated at common score values. The different traditions in how the curves are created arises out of the goals of the communities. Information retrieval evaluations have typically considered just the rankings; TDT was conceived with score normalization as a critical component.

3.3 Tracking evaluation

Within TDT, tracking is evaluated by considering how well a system's decisions about the test stories matches the truth. The evaluation corpus consists of every story that follows the N_t^{th} training story.[4] That means that every topic has a slightly different evaluation subset of the corpus.

As mentioned above, the system is expected to output a score for every evaluation story, indicating the system's confidence that the story is on topic. The scores are *required* to be comparable *across* topics, and a global threshold must be used to make a hard "yes or no" decision about every story.

The formal evaluation measure for tracking is a cost measure calculated at the hard decision point and averaged over all topics. In addition, the DET curve shows the tradeoffs between miss and false alarm rates, again averaged over all topics. Note that the DET curve is plotted without any cross-topic score normalization, so scores that are not comparable across topics are likely to reduce effectiveness.

3.4 Detection evaluation

Detection is evaluated by comparing the clusters generated by the system to the truth clusters. Each topic in the evaluation data should ideally be represented by a single cluster. The evaluation proceeds by creating a mapping between clusters and true topics, selecting the mapping that minimizes the detection cost function for each topic (i.e., the most generous mapping of this system's output). Once the mapping has been created, it is possible to calculate miss and false alarm rates for each of the topics that are known (in our case, the 60 evaluation topics). It is obviously not possible to evaluate the quality of clusters that are not associated with a known topic, so their quality is ignored and they do not affect the performance of a system (although systems are required to generate them).

The formal evaluation measure for detection is the cost measure averaged over all topics. In addition, a DET "cloud," is created that has one point for every topic, showing the error tradeoff for that topic. The DET plot also includes the average error tradeoff over all topics. Figure 3 shows a sample DET cloud.

It is not possible for the detection evaluation to include a DET tradeoff line because the system does not output the similarities of each story to *all* topics—only to the topic to which it is assigned. We believe this situation, unavoidable in the current evaluation paradigm, is a major handicap of detection evaluation. We postulate that converting detection to tracking may help with the problem.

4 Experiments

We will use TDT system output to examine the relationship between tracking and detection. In this section, we describe how the experiments were run. All runs are based upon parameter settings chosen for the TDT 2000 evaluation workshop [NIST, 2000b]. That system and its parameter styles is the result of several years of involvement in TDT [Papka, 1999, Allan et al., 2000d, Allan et al., 2000a].

[4]It actually starts slightly later than that, because stories are delivered in sets called a "file." Tracking evaluation formally begins at the start of the next file.

4.1 Corpora

All experiments were run using the TDT-3 corpus of news stories recorded October through December of 1998. The corpus includes approximately 34,600 stories from eight English sources and 30,000 stories from three Mandarin news sources (we used the supplied SYSTRAN-generated translations of the Mandarin stories).

As part of the TDT 1999 and TDT 2000 evaluations, the Linguistic Data Consortium (LDC) identified 120 news topics in the corpus and determined the subset of stories that discuss each of those topics. Because of the way they were created, there is high confidence that almost all on-topic stories have been identified. For these experiments, we use the 60 topics chosen from those 120 to be used as training in TDT 2001 (numbered in the range 30001 to 31060).

Training to set thresholds and select other system parameters was done using the 60,000-story TDT-2 corpus and its 200 topics [Cieri et al., 1999].

4.2 Tracking runs

The core of our TDT system uses a vector model for representing stories—i.e., we represent each story as a vector in term-space, where coordinates represent the frequency of a particular term in a story. Terms (or features) of each vector are single words, reduced to their root form by a dictionary-based stemmer. This system is based on one that was originally developed for the 1999 summer workshop at Johns Hopkins University's Center for Language and Speech Processing.[Allan et al., 1999]

For tracking, we group the N_t training stories into a cluster that represents the topic being discussed. The cluster is represented by a *centroid*, which is an average of the vector representatives of the training stories. Vectors were built from the most frequent 1,000 terms of each story (large enough that it is usually all terms in the story).

Incoming stories are compared to the centroid of the cluster, and if the similarity of the story to the cluster exceeds a threshold, $\theta_{match} = 0.07$, we declare the story "on-topic" for the cluster. We used a cosine similarity function to compare vectors. The measure is simply an inner product of two vectors, where each vector is normalized to unit length. It represents the cosine of the angle between the two vectors d and q.

$$\frac{\sum q_i d_i}{\sqrt{\sum q_i^2 \sum d_i^2}}$$

(If $\vec{q}$ and $\vec{d}$ have unit length, the denominator is 1.0 and the angle is calculated by a simple dot product.) Vectors were weighted by the raw tf of the feature times an InQuery IDF component, $\frac{\log(N/df)}{\log(N+1)}$, where N is the number of stories seen to date and df is the total number of those that contain the feature.

4.3 Detection runs

For the detection task, we used a very similar approach, also taken from our TDT 2000 workshop run. Stories were represented by vectors using the top 1,000 terms and compared using the cosine measure. When a new story arrived, it was compared against *all* previously seen stories. If its similarity with the strongest match was above 0.20 (a threshold selected from training data), then the new story was added to the same cluster as the top-matching story. If its similarity was below the threshold, the arriving story was declared to be the start of a new cluster.

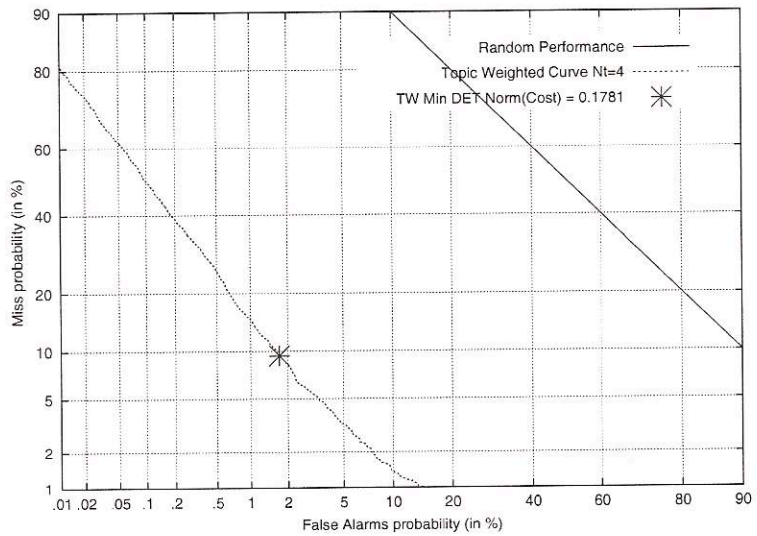

Figure 1: DET plot for tracking run with $N_t = 4$. This graph shows the baseline performance for an actual tracking system.

Note that although tracking and detection are very similar tasks and we trained to minimize the same cost function (normalized $\text{Cost}_{\text{TDT2000}}$), the thresholds for detection and tracking are substantially different.

5 Results

In this section, we discuss the results of the detection and tracking systems, and compare them by treating the former as a different type of tracking. We show that although the two tasks are fundamentally related and measured by the same cost function, their differing evaluation approaches yield conflicting parameter choices.

5.1 Tracking evaluation

The results of our tracking run [NIST, 2000b] at $N_t = 4$ are shown in the DET curve of Figure 1. The false alarm rate runs along the x-axis and the miss rate is on the y-axis. Recall that better systems result in curves closer to the origin (low error rates). This graph shows the effectiveness of tracking running from 0.01% false alarm at an 80% miss rate down through about 14% false alarm at 1% miss rate. The marked point on the curve represents the point where normalized tracking cost is minimized. That is, where:

$$\text{Norm}(\text{Cost}_{\text{TDT2000}}) = 0.02 \cdot \text{miss-rate} + 0.002 \cdot \text{fa-rate}/0.002$$

is at a minimum (0.1781). The system-selected threshold (not shown) resulted in a tradeoff of approximately P(Miss)=6% and P(FA)=3%, or a normalized tracking cost of 0.1917.

For comparison, Figure 2 shows the recall and precision of the same system. Note that recall/precision graphs are created by interpolation across topics and then averaging, so score normalization is not an issue. For researchers more used to the recall/precision graphs, it is clear that TDT tracking effectiveness is higher

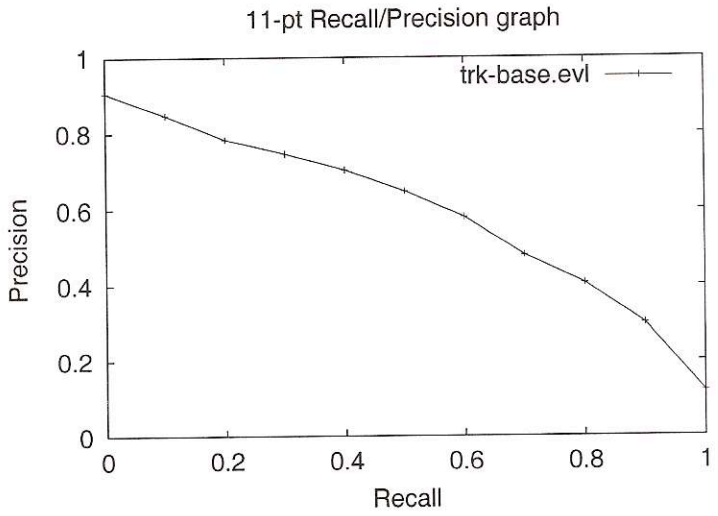

Figure 2: Recall/precision tradeoff plot for the tracking run ($N_t = 4$) depicted in Figure 1. This graph allows comparison between DET plots and recall/precision tradeoff plots.

than that typically found in document retrieval tasks. That graph corresponds to an average precision of 59%.

5.2 Detection evaluation

Figure 3 shows the performance of our detection system using a DET "cloud." The plus signs represent the miss and false alarm rate values for *each* of the evaluation topics. The stars on the x- and y-axes represent points where the miss or false alarm rate is off the edge of the graph (the Gaussian scale means that a rate of 0.0, for example, is infinitely far to the left or bottom). The "×" mark at P(Miss)=24% and P(fa)=1% represents the average score by topic, and corresponds to a normalized detection cost of 0.2859. This result is immediately surprising because it is so much worse than the tracking score, even though it is evaluated on the same 60 topics and with the same cost function. In the next section, we will convert the detection task into a tracking task so that the results are more directly comparable.

It is also not possible in Figure 3 to estimate the impact that changing thresholds would have on the accuracy of the systems. Changing the threshold when running the system would cause different stories to be added to the clusters, would result in different mappings between clusters and truth topics, and would end with a non-comparable DET "cloud."

5.3 Detection as tracking

As discussed in Section 2.3, it is possible to view the detection task as parallel tracking of multiple topics. To do that, we converted detection system output into tracking output for $N_t = 1$ as follows:

1. Assume topic T has $N_t = 1$ training stories, S_1.

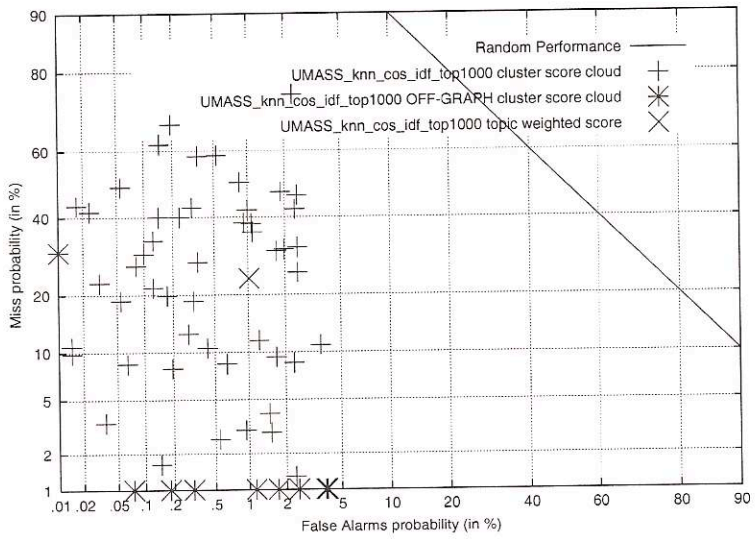

Figure 3: *Detection* evaluation DET plot ("cloud") for detection run. Each topic in the detection system's output is represented by a single point on the plot.

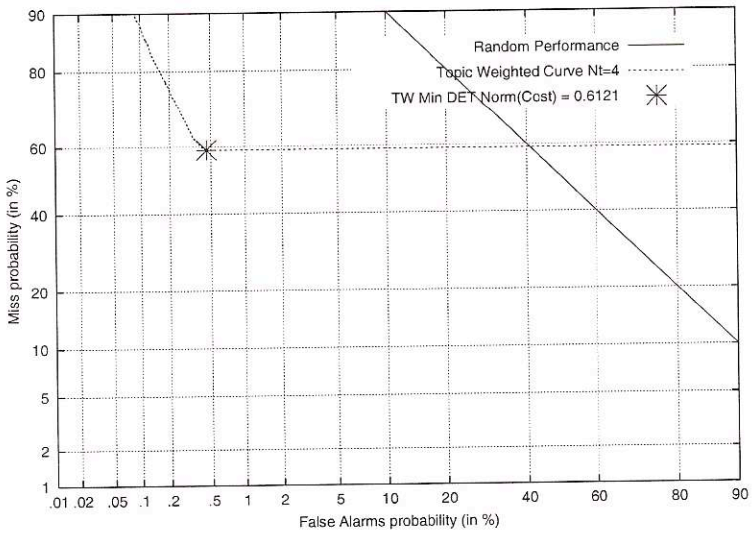

Figure 4: *Tracking* evaluation DET plot ($N_t = 1$) for detection run. This plot represents the detection system evaluated in Figure 3, but where the output is viewed as tracking output.

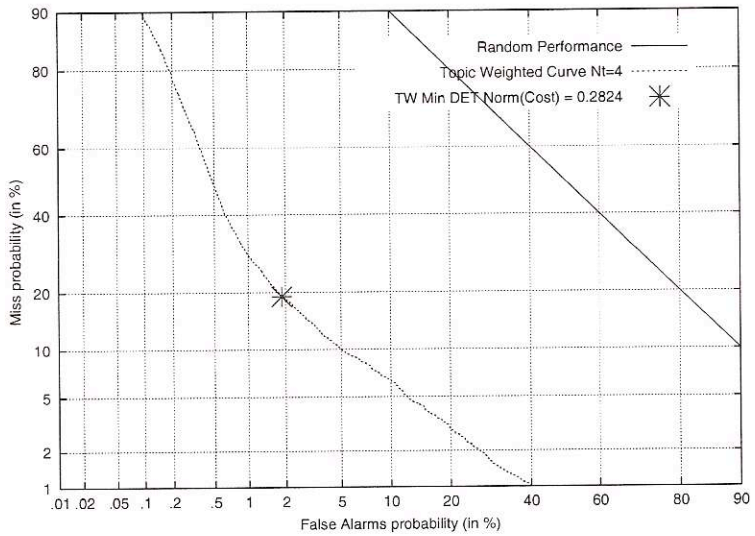

Figure 5: *Tracking* evaluation DET plot for detection run. This plot represents the same system as Figure 4, but with the hard decision point removed.

2. Find the cluster C_1 that story S_1 is assigned by the detection system. Note that S_1 might have started C_1 or it might have joined the cluster already in existence.

3. For every story S_j that occurs after the training story S_1 (i.e., the evaluation set for topic T), we assign a tracking score. If S_j is placed in cluster C_1, we use the detection score for S_j as its tracking score. The detection score represents the system's confidence that S_j should be assigned to cluster C_1—i.e, the same cluster as the training story S_1.

4. If S_j is not placed in C_1, we do not have score information for that story with respect to C_1, so we assign it a tracking score of zero.

Figure 4 shows the resulting tracking evaluation DET curve. It has a typical shape on the left, but at roughly the 60% miss rate, the plot flattens and the miss rate does not drop. This effect is the result of having scores only for stories that were successfully assigned to the training story's topic. What the curve shows is that if the threshold for assigning stories to clusters were raised from the chosen 0.20 (see Section 4.3), the error tradeoff would be as shown. However, the limitations of the conversion process described above means that the system has error rates for the 0.20 threshold, but not for 0.19, 0.18, etc. It can only jump to the error rates at a threshold of 0.0, where the miss rate goes to 0% and the false alarm rate is 100%. (The evaluation software represents that condition with a line parallel to the axis.)

To address this problem and get a better sense of how detection and tracking compare, we modified our detection system so that it emitted detection scores with respect to any cluster that contained a training story for any topic. That means that step (4) above effectively no longer exists because we have a score for every story with respect to that cluster, regardless of which cluster it was actually assigned to. Figure 5 shows the resulting DET plot, where the curve now extends the full length of the graph.

At this point, we can see that the optimal tracking point was far from the threshold used in the detection system, even though the detection performance was respectable (in comparison to other systems that

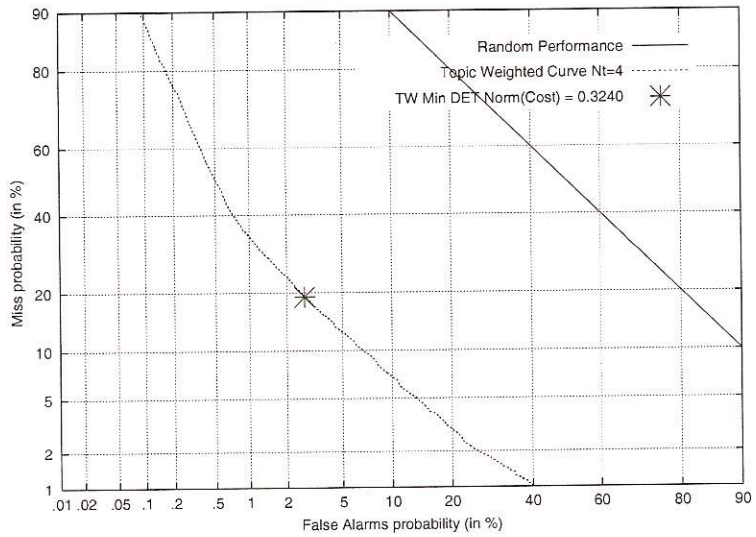

Figure 6: Tracking evaluation DET plot for detection run. Score is the *average* score for clusters containing a training story.

participated in the TDT 2000 evaluation). However, lowering the threshold for detection did not improve the results for that task, even though this graphs suggests it would. For example, dropping the threshold from 0.20 to 0.18 caused a 10% *increase* in cost, not the drop that the minimum cost points suggests would happen.

To see whether or not the value of N_t would affect the conversion from detection to tracking, we modified the approach as follows. We used the modified system above so that we have a score for every cluster associated with a training story.

1. Assume topic T has $N_t = 4$ training stories, S_1, S_2, S_3, and S_4.
2. Find the cluster C_i that story S_i is assigned by the detection system. Note that S_i might have started C_i or it might have joined the cluster already in existence. Also note that it is possible that $C_i = C_j$ even when $i \neq j$.
3. For every story S_j that occurs after the training story S_4, we assign a tracking score. We use the average of score(S_j, C_i), assigning the tracked story S_j its average connection to any of the training stories.

Figure 6 shows the effect of mapping to multiple training stories. There is very little difference between that graph and Figure 5, although the original ($N_t = 1$) is slightly better. This is surprising because tracking performance at $N_t = 4$ is substantially better than at $N_t = 1$. It appears that averaging the location is not an appropriate way to convert to tracking when $N_t > 1$ because it adds stories from multiple clusters, thereby greatly increasing the number of false alarms.

The difference is not large, probably because 37 of the 60 evaluation topics had all $N_t = 4$ training stories in the same cluster. Another 9 had three in one cluster, two topics had a 2–2 split, eight had a 2–1–1 split into three clusters, and four topics had the training stories separated into four clusters. Because in over 75%

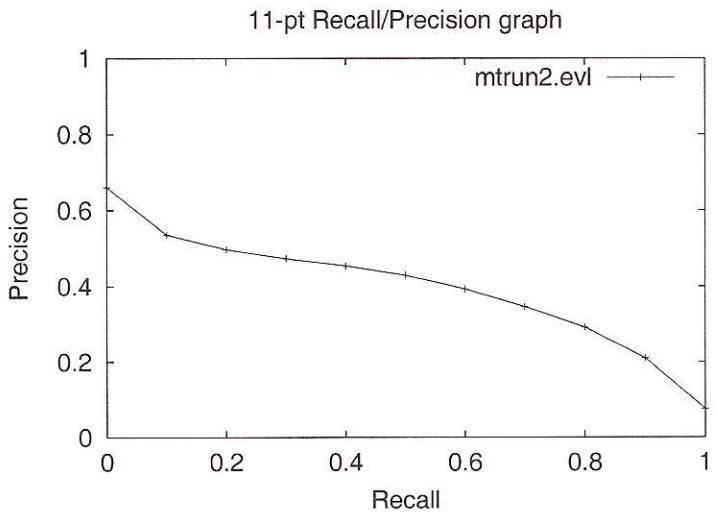

Figure 7: Recall/precision tradeoff graph for detection run viewed as tracking.

of the cases almost all stories were in the same cluster, there was no or little change in the score of a tracked story compared to that cluster. For the same reason, the plot was nearly identical when a story was assigned the *maximum* similarity to C_1 through C_4, rather than the average score.

As a final comparison, Figure 7 shows the graph of Figure 6 converted to a recall/precision graph measuring tracking performance. This graph, compared to Figure 2, highlights the poor tracking performance achieved by the detection system.

6 Conclusion and future work

There is a general sense in the TDT research community that one approach is sufficient for each of the various filtering tasks. Specifically, given the official evaluation paradigm, that tracking and detection can be approximated using the same techniques. To be sure, researchers have employed different approaches to the tasks, with decision trees [Yang et al., 1999] being one of the obviously distinct approaches. However, most research has employed the same core statistical model for detection and for tracking, be that language modeling, k-NN, or a straightforward vector model approaches. The fundamental task then becomes fitting parameters (value of k, term weighting function, statistical estimators, smoothing functions, etc.), using simple exhaustive sweeps [NIST, 2000b], or complex and rigorous machine learning approaches [Yang et al., 2000] for the different tasks.

We are not surprised that the optimal threshold for the detection task does not correspond to the best threshold for the tracking task: the tasks are different and it is to be expected that ideal parameters and thresholds would shift. However, we have shown that when the detection task is treated as a tracking task, the effectiveness is significantly worse than "true" tracking, even though the core approaches to the tasks are identical.

The reason for this discrepancy is that the two tasks actually measure different things. The goal of a tracking system is to put all follow-up discussion of a topic in the same category as the first several stories.

However, the goal of a detection system is to keep the bulk of the stories on a topic in the same cluster.

A tracking system that treats, say, 80% of the later stories as being on a separate topic, and therefore fails to track them, will do badly. However, if the system views that 80% of the topic as a single (but different) topic, it will do very well as a detection system. It will have clustered the bulk of the topic into a single "bin" even though it missed the first several stories. The mapping from clusters to truth topics will select the best cluster, even though that is not the cluster that a tracking system would have selected.

As a result, even though detection and tracking are very similar, and even though a high-accuracy tracking system would come close to solving the detection task, mistakes in tracking do not necessarily translate into mistakes in detection. Approaches that might utterly fail in tracking still have the *potential* to be useful in detection. It seems less likely that a poor-quality detection system would work well for tracking, since poor detection generally means that the topic was highly fragmented into many clusters. That suggests the likelihood that a corresponding tracking system would also fragment the topic, and fail to track most of it.

It is our contention that although traditional filtering techniques have much to offer tasks such as TDT cluster detection, the slightly narrower domain and the different evaluation paradigm allow and perhaps require a wider range of approaches. We are investigating a broader range of models for the detection task, including more elaborate modeling of the underlying events that make up topics, as well as the human-generated "rules of interpretation" that describe the boundaries of what is and is not on topic.

We are also investigating alternate evaluation approaches for the clustering task. The mapping between hypothesis clusters and the truth topics is a major problem with the evaluation. A system that breaks a topic into two large clusters is no better, by the TDT measure, than a system that breaks the topic into one large cluster and countless singletons. There have been efforts to address that in TDT, including the YDZ measure. However, none of those has been satisfying to the TDT research community.

We continue to be interested in the relationships between various tasks, believing that comparing the tasks can illuminate aspects of each that were previously non-obvious. We have previously shown that the relationship between tracking and first story detection discourages basing systems to solve the latter on technology to address the former [Allan et al., 2000b, Allan et al., 2000c]. Here we show that the fundamental differences in evaluation paradigm make two strongly related tasks—tracking and detection, a form of multi-topic parallel tracking—not as similar as they appear at first. This suggests that different approaches to the cluster detection are warranted and should be encouraged.

Acknowledgments

I am grateful to Victor Lavrenko for his help modifying our TDT system and evaluation software for this study.

This work was supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912, and in part by the Air Force Office of Scientific Research under grant number F49620-99-1-0138. The opinions, views, findings, and conclusions contained in this material are those of the author and do not necessarily reflect the position or policy of the Government and no official endorsement should be inferred.

References

[Allan et al., 1998a] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. (1998a). Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218.

[Allan et al., 2001] Allan, J., Carbonell, J., and J. Yamron, e., editors (2001). *Topic Detection and Tracking: Event-Based Information Organization.* Kluwer Academic Press. In process.

[Allan et al., 1999] Allan, J., Jin, H., Rajman, M., Wayne, C., Gildea, D., Lavrenko, V., Hoberman, R., and Caputo, D. (1999). Topic-based novelty detection: 1999 summer workshop at CLSP, final report. Available at http://www.clsp.jhu.edu/ws99/tdt.

[Allan et al., 2000a] Allan, J., Lavrenko, V., Frey, D., and Khandelwal, V. (November 2000a). UMass at TDT 2000. In *Proceedings of the TDT Workshop.* Unpublished.

[Allan et al., 2000b] Allan, J., Lavrenko, V., and Jin, H. (2000b). Comparing effectiveness in TDT and IR. Technical Report IR-197, CIIR, Department of Computer Science, University of Massachusetts, Amherst.

[Allan et al., 2000c] Allan, J., Lavrenko, V., and Jin, H. (2000c). First story detection in TDT is hard. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*, pages 374–381.

[Allan et al., 2000d] Allan, J., Lavrenko, V., Malin, D., and Swan, R. (March 2000d). Detections, bounds, and timelines: UMass and TDT-3. In *Proceedings of the TDT Workshop.* Unpublished.

[Allan et al., 1998b] Allan, J., Papka, R., and Lavrenko, V. (1998b). On-line new event detection and tracking. In *Proceedings of ACM SIGIR*, pages 37–45.

[Cieri et al., 1999] Cieri, C., Graff, D., Liberman, M., Martey, N., and Strassel, S. (1999). The TDT-2 text and speech corpus. In *Proceedings of the DARPA Broadcast News Workshop*, pages 57–60.

[DARPA, 1999] DARPA, editor (1999). *Proceedings of the DARPA Broadcast news Workshop*, Herndon, Virginia.

[Martin et al., 1997] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and ybocki, M. (1997). The DET curve in assessment of detection task performance. In *Proceedings of EuroSpeech'97*, pages 1895–1898.

[NIST, 2000a] NIST (2000a). Proceedings of the TDT 1999 workshop. Notebook publication for participants only.

[NIST, 2000b] NIST (2000b). Proceedings of the TDT 2000 workshop. Notebook publication for participants only.

[Papka, 1999] Papka, R. (1999). *On-line New Event Detection, Clustering, and Tracking.* PhD thesis, Department of Computer Science, University of Massachussetts.

[Robertson and Hull, 2001] Robertson, S. and Hull, D. A. (2001). The TREC-9 filtering track final report. In *Proceedings of TREC-9.* Forthcoming; see also http://trec.nist.gov.