Event Tracking

James Allan, Victor Lavrenko, and Ron Papka

Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts
Amherst, MA 01003

Abstract

This paper introduces *Event Tracking*, a new application of Information Retrieval technology with interesting research and evaluation questions. We describe the problem, a pilot corpus of news stories that was constructed for experimental studies, and a "rolling" evaluation strategy that uses different segments of the corpus for each query. As part of a preliminary evaluation on a small pilot study corpus, we show that simple feature extraction methods can generate moderately accurate event representations, but that some classes of events are handled much better by letting the representation adapt to the event's evolution in the news. Event tracking is part of the Topic Detection and Tracking initiative.

1 Introduction

Event Tracking is the task of monitoring a stream of news stories to find those that discuss the same event as the one covered in a few sample stories. For example, having read one or two stories about a bombing, a user might tag those stories and ask that the system notify him or her when new stories on the same event are broadcast. Event Tracking is a close cousin to the Information Retrieval (IR) problems of Information Filtering and Routing. The major differences are two-fold:

1. Event Tracking is restricted to the domain of news. It could be extended somewhat to monitor less "pure" sources, but news is an ideal medium for finding coverage of events. Filtering and Routing, on the other hand, are generally applied to unrestricted corpora covering arbitrary topics.
2. The "query" in Event Tracking refers to an event, something in the real world that happened, and is specified using a few examples. By contrast, IR queries are generally at the broader level of topic, and are described directly by the user.

One of the problems within the Information Filtering research community has been finding a "realistic" application of filtering, with a useful corpus and an appropriate evaluation methodology.[12] The problem is that Information Filtering is a technology of such broad applicability that slightly different notions of its purpose lead to substantially different evaluation measures.

Event Tracking is a form of Information Filtering, so it provides an alternate venue for researching open questions. On the other hand, this new task is more narrowly defined, meaning that evaluation measures can more easily be agreed upon. Further, the shift of focus into a somewhat more specific domain opens the doors to the possibility of applying special purpose techniques (e.g., natural language processing) to improve effectiveness.

In the following, we first describe the problem and history of Event Tracking in more detail. In Section 3 we describe the evaluation corpus and evaluation methodology used in this study, including an introduction to the "DET curve." Section 4 explores the effectiveness of simple filtering techniques for the Event Tracking

University of Massachusetts, Computer Science Department, CIIR Technical Report IR-128, January 1998.

problem, and Section 5 extends and improves those methods by using "adaptive" tracking. Finally, in Section 6 we discuss our conclusions, the state of the art for this problem, and talk about future directions.

We feel it is important to stress that this work was carried out as part of a pilot study with a relatively small corpus (by the standards of Information Retrieval). The results should therefore be taken as suggestive and not conclusive.

2 History and Definition

The Event Tracking problem is part of a broader initiative called Topic Detection and Tracking (TDT). The domain of TDT's interest is all broadcast news—i.e., written and spoken news stories in multiple languages. As such, the problem is substantially broader than the work reported in this study, encompassing automatic speech-to-text efforts, finding the boundaries between news stories for archival and presentation purposes, locating new events within the stream, tracking located events, and doing all of that in a multi-lingual environment with degraded information. As its name implies, TDT is also ultimately concerned with ways of organizing information that are broader than "events," but the initial work has focused on the more limited setting.

The TDT tasks and evaluation approaches were developed by a joint effort between DARPA, the University of Massachusetts' Center for Intelligent Information Retrieval, Carnegie Mellon's Language Technology Institute, and Dragon Systems. A year-long pilot study was undertaken to define the problem clearly, develop a test bed for research, and evaluate the ability of current technologies to address the problem. Results of that study were reported at a workshop in October of 1997 and a final report was made at a related workshop[1] The groups involved in the tasks found that current methods (discussed briefly in Section 4.4 below) are capable of providing adequate performance for detection and tracking of events, but that there is a high enough failure rate to warrant significant research into how the state of the art can be advanced. As the research is broadened to the larger TDT scope, the unresolved questions become more troublesome.

We wish to make it clear that the corpus and evaluation methodology that were devised in the TDT study were a joint effort by four groups. The research results on Event Tracking reported in this study are our own; the framework for the work is only partly ours.

2.1 Handling news

The tasks defined within TDT, Event Tracking among them, appear to be new within the research community. Some efforts have been made to classify news stories into broad topic areas automatically using nearest neighbor matching[15], pattern matching[9], or frame-based representations[7]. For the most part, those techniques are intended to match stories against a set of *topic* labels that are known *a priori*. Event Tracking requires finding stories that discuss an event that may not match any already known class of events.

Because it connects stories together by the driving event, Event Tracking shares some similarities with automatic hypertext creation. There has been work on making such links in a general setting[3] and in electronic mail[13], but that work did not touch upon events.

Some recent work that associates news photographs with stories about the picture[6] is very similar in spirit to Event Tracking. Their stated interest is in linking stories that discuss the same event, though they work solely on the problem of linking photographs and stories. As we will do in our work, they represented stories and photo captions by sets of features. They found that proper names are very useful for linking.

3 Evaluation corpus and methodology

An important task of the TDT pilot study was the creation of an appropriate test corpus and a useful approach to evaluation of the problem. The goals of creating the corpus and evaluation methodology were two-fold: (1) to make strides toward a solid definition of "event" and (2) to evaluate how well "state of the art" approaches could address the TDT tasks. Because no *a priori* agreeable answer to point (1) could be found, one hope of this work is than an interactive process of partial definition, evaluation, better definition, evaluation, and so on, will eventually result in a satisfactory conclusion.

To simplify the problem slightly for the pilot study, we generally ignored issues of degraded text coming from speech recordings, and used written newswire sources and human-transcribed stories from broadcast news. The resulting TDT corpus includes 15,863 news stories spanning July 1, 1994, through June 30, 1995. Half of the stories are randomly chosen Reuters news articles from that period; the other half are transcripts of several CNN broadcast news shows during the same period. The stories are assigned an ordering that represents the order that they appeared in the news.

The corpus also includes relevance judgments for a set of 25 events. Some events (e.g., the Oklahoma City bombing or the earthquake in Kobe, Japan) were disasters or crimes that occurred in the news and were unexpected. Others are stories that build up to an anticipated event (e.g., the collision of a comet into Jupiter, the appointment of U. S. Supreme Court Justice Breyer). The events were chosen to represent a range of events that seemed "interesting," and such that there would be a fair number of stories on each event in the corpus. Recall that there is no agreed-upon definition of "event" *per se*, so the 25 events were also chosen to cover a range of types of events that were generally acceptable.

To provide a high-quality evaluation setting, *every* story in the corpus was judged with respect to *every* event. The judgments were made by two sets of assessors and any conflicts were reconciled by a third. For each of the 25 events, each of the stories was assigned a judgment on a ternary scale: about the event, not about the event, or mentioning the event but only briefly in a story that is generally not about the event. The exhaustive judgments of this corpus are in contrast to more common pooled strategies.[20] An unfortunate side-effect of requiring exhaustive judgments is that the cost of creating them limits the size of the corpus.

The TDT corpus and relevance judgments are available from the Linguistic Data Consortium. The LDC is currently creating a second and larger TDT corpus that includes a broader range of sources, as well as the audio stream and closed captioning for all broadcast sources, and a larger number of judged events.

3.1 Evaluation methodology

The TDT evaluation approach is different than the more established TREC filtering task. The latter provides a large amount of training data with queries and relevance judgments, and requires that sites generate filtering queries that will work on a test set provided later. In the TREC-6 filtering track,[21] the training data includes anywhere from 6 to 887 relevant documents, with a mean of 123 (the routing track had between 8 and 2,431 relevant documents with a mean of 576). Although there are settings where that much training information is possible, it is difficult to argue that the setting is commonplace.

For the Event Tracking task, on the other hand, we are interested in substantially smaller numbers of training stories—in fact, we are interested in how few stories are needed for successful tracking. However, a more important problem for this task is that to model a real world setting, the tracking needs to begin as soon as possible after the training stories are "presented." Consider the case of tracking a bombing event: it is not interesting to evaluate a tracking system on news that is reported weeks after the event—the goal of the system is to begin tracking *immediately*. Unfortunately, events occur at different times, meaning that it is nearly impossible to use the same training and test set for each event.

For those reasons, the TDT corpus is split into training and test information at a different point for each event. If the system is being evaluated for 4 training stories, then the training corpus is all stories up to and including the fourth training story and the test corpus is the remainder of the corpus. Note that this also means that different numbers of training stories create different training and test corpora.

In this study, we let the number of training stories, N_t, take on values 1, 2, 4, 8, and 16. If an event has fewer than N_t training stories it is neither tracked nor evaluated at that N_t value.[1] To compare system performance across N_t values, the system is trained on N_t stories, but always evaluated on the $N_t = 16$ test set—i.e., its performance on the stories between the N_t^{th} and the 16^{th} training story is ignored. Ten of the 25 events have fewer than 16 training stories, so cross-N_t comparisons are done using only 15 events.

The effect of this per-event, per-N_t separation of the corpus into training and test data is to create a "rolling" evaluation corpus. A tracking system will be testing some events while it is simultaneously training others.

[1] Note that this means that for larger values of N_t, only heavily reported events will be evaluated. This effect means that results for one value of N_t should not be assumed valid for other values.

3.2 Evaluation measures

Information Retrieval systems are generally evaluated on the basis of ranked recall and precision,[10, 17] though numerous other measures have been proposed.[8, 19] Information Filtering systems are evaluated on a wider range of measures, including set-based measures and various utility measures, though no particular measure has settled out as preferred.[12]

In the TDT setting, we have chosen to measure a system's effectiveness primarily by the miss and false alarm (fallout) rates. The major reason for choosing these is a perception of the problem as being a *detection* task rather than a ranking task: a system needs to indicate whether or not a story is on the event being tracked, not provide a ranked list of stories that might discuss the event. Unfortunately, although it is fairly straightforward for systems to generate ranked lists of stories and to provide a score that creates that ranking, it is extremely difficult to determine a good score that can be used as a threshold. An ideal system might output a score that corresponds to the probability that the story discusses the event; ideal systems do not exist.

In this work, we skirt the threshold issue by using a Detection Error Tradeoff curve[14] to show how false alarm and miss rates vary with respect to each other at various threshold values. Figure 1 shows examples of DET plots showing curves for several different runs. The curves are plotted such that if the errors exhibit a normal distribution, they will result in a straight line. A perfect system would have zero misses and zero false alarms, and would have a "curve" at the origin; curves closer to the origin are generally better, though there may be applications that require good performance at particular false alarm or miss rates. For most applications, the left-hand side of the DET curve (low false alarm rate) is probably the most interesting. For our particular study, a false alarm rate of 1% means that as many as 158 stories per event would have been incorrectly tracked.

The DET graph is analogous to a recall/precision graph, except that "good" is in the opposite direction. Both graphs provide a means for comparing system performance across a wide range of error tradeoffs. Both allow a user to understand the tradeoff between improving one measure and worsening the other.

4 Simple tracking methods

This section discusses the effectiveness of some simple approaches to event tracking, based primarily on Information Filtering methods. We find effective tracking queries just by using a small number of features appropriately weighted. Larger numbers of features tend not to work well, in contrast to past TREC routing results.[4]

4.1 Tracking approach

For the tracking runs in this study, we created two databases. The first ("tdt") consists of only the TDT corpus. The second ("past") adds some additional training data to the start of the corpus so that "collection" statistics such as idf are more likely to be meaningful.

We represent stories by vectors of features. The features were found by applying a shallow tagger[22] to the stories and selecting all nouns, verbs, adjectives, and numbers. Names of countries, states, and large cities were treated as single features by the tagger even if they are multiword. There was no stopword list, but most common stopwords do not fall into the parts of speech used. The features were stemmed.

Queries are represented by a similar vector of features. Queries and stories are compared by:

$$\begin{aligned} sim(Q,D) &= \frac{\sum_{i=1}^{N} q_i \cdot d_i}{\sum_{i=1}^{N} q_i} \\ d_i &= \frac{tf}{tf+2} \cdot (1 - \log_N df_i) \end{aligned}$$

where q_i is the weight of feature i in the query, d_i is the weight in the story, tf is the number of times the feature occurs in the story, df_i is the total number of the times the feature occurs in the collection, and N is the number of stories in the collection. (This weighting function is a simplification of the more complex weighting scheme currently used in InQuery[21]; it assumes that all stories are of roughly the same length, that the collection is never empty, and that df_i is not zero.)

In order that the tracking system model a real setting, the "collection" information cannot be known in advance. For that reason, the corpus-wide parts of the weighting function (N and df_i) are "values to date" and represent information for the portion of the corpus (training *and* testing) that has already been seen. That is, at the moment a query is created, the *current* values of N and df_i are used (the query is not re-adjusted) and at any point that a story is converted to a vector, the values of N and df_i up to and including that story are used.

A final step in the scoring process normalizes scores across all events. The comparison function above results in a ranking of stories where the higher in the rank the more likely a story is to discuss the event in question. However, a score of 0.45 could mean "very likely to match" for one query and "very unlikely" for another query. Our goal is to normalize the scores so that 0.45 (and every other score) has roughly the same meaning no matter what query and story are being compared. This should result in more "meaningful" scores for stories and more appropriately matches the assumptions behind the DET curve discussed above.

To normalize scores, we calculate the similarity of the query against the N_t training stories and find the average of those similarities. That average value is used as a normalization factor, and all scores (for that event) are divided by it. As a result, although scores can range from zero through well above 1.0, a particularly "good" story (for any event) should score 1.0 or higher. That is, its unnormalized score will be near those of the N_t training stories, so dividing by that average score, will result in a normalized score near 1.0. The interpretation of 1.0 as "very like the training stories" is more likely to be true across all events than before normalization.[2]

Our second collection ("past") prepends the TDT stories with 31,188 CNN news stories from January 1, 1993, through June 30, 1994—i.e., the 18 months immediately before the TDT corpus. The intent of providing that data is to give the collection-so-far statistics a more meaningful basis, although there is evidence that doing so is not critical.[5]

4.2 Using common words

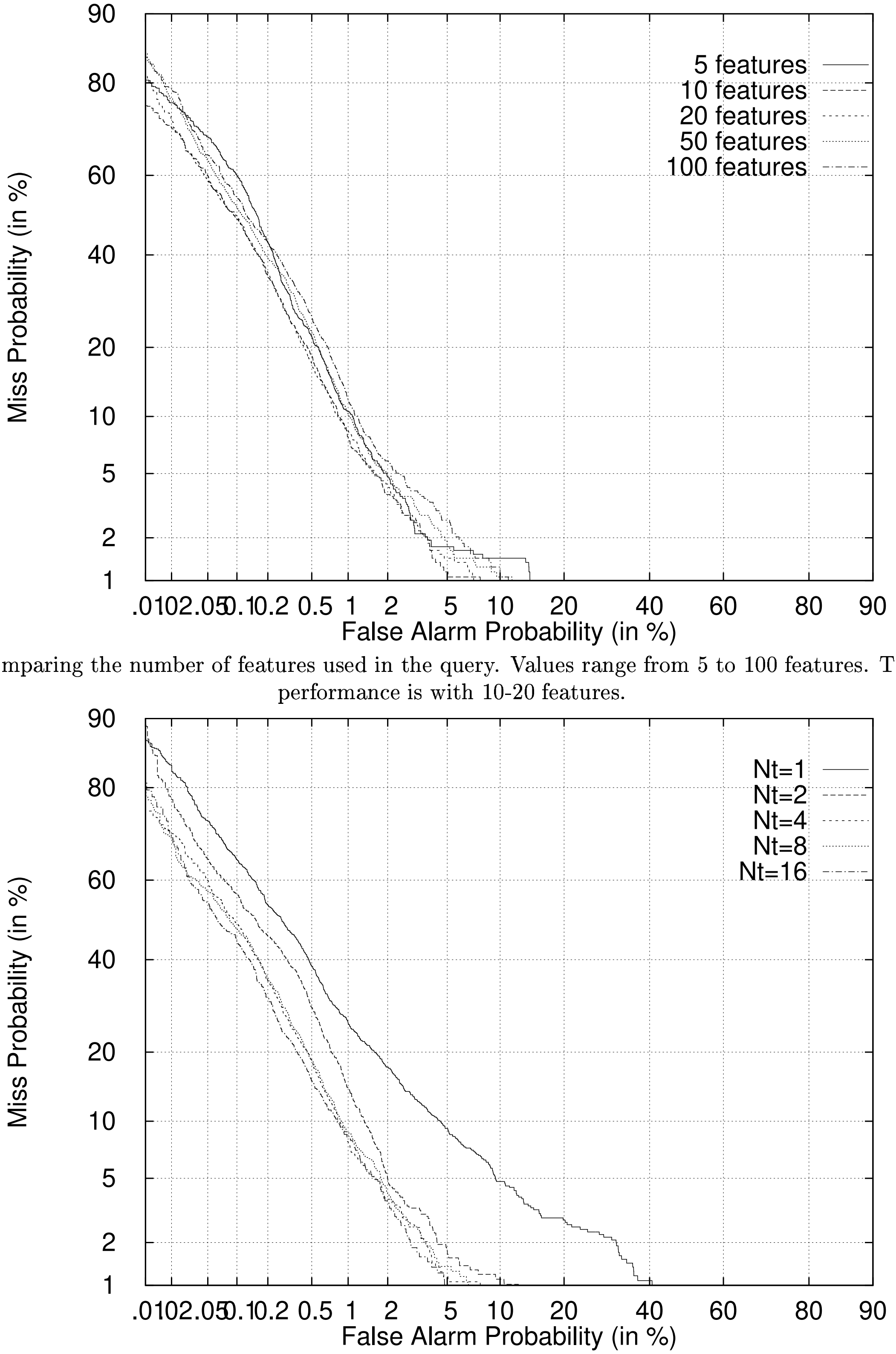
The simplest approach to tracking is to select useful words or phrases from the N_t training stories and use those to form a query and a threshold for matching the query. As in any filtering task, all subsequent stories are compared against the query and, if the match is above the threshold, the story is selected—here, it is assumed to be about the same event.

We used the top n most commonly occurring features in the N_t training stories, with weight equal to the number of times the feature occurred in those stories multiplied by its incremental idf value (set after the N_t^{th} story). Figure 1a shows the effect of various values of n on the tracking effectiveness of the 25 events at $N_t = 4$. As the DET curve shows, the performance is similar, though larger queries are generally less effective; the optimal values appear to be 10-20 features. We suspect a small number of features is sufficient because news reporting tends to focus on important words and phrases to distinguish between different news events and because some stories occur to people or in places that are normally not in the news: capturing the one or two "killer features" is sufficient to track the event with high accuracy. We have not investigated this issue in depth.

The number of terms in the query is not particularly important at any value of N_t. Figure 1b shows 10-feature queries at several values of N_t. The curves show that more training helps the performance, but that by the time there are four sample stories, adding more provides little help. This rapid peaking of effectiveness is similar to that observed in the TREC routing tasks.[2]

To consider the impact of corpus-wide statistics on the effectiveness of Event Tracking, we generated two sets of queries, each with 10 features. The first set had the features weighted by the number of times that the feature appeared in the relevant training stories (rcf). The second set multiplied that value by the incremental idf value (rcfidf). The queries were run against both of our test databases: tdt and past. Figure 2 shows the impact of the runs. It is unfortunately difficult to see, but adding the idf information to

[2] We have not yet performed any statistical analysis of the normalization to know details of its effect. It does improve the detection error tradeoff as represented by the DET curve, so we believe that we are achieving something useful for situations where that sort of measure is important.



(a) Comparing the number of features used in the query. Values range from 5 to 100 features. The best performance is with 10-20 features.

(b) Comparing values of N_t. Once N_t reaches 4, adding more stories for training is only marginally helpful.

Figure 1: Comparing number of features and number of training instances.

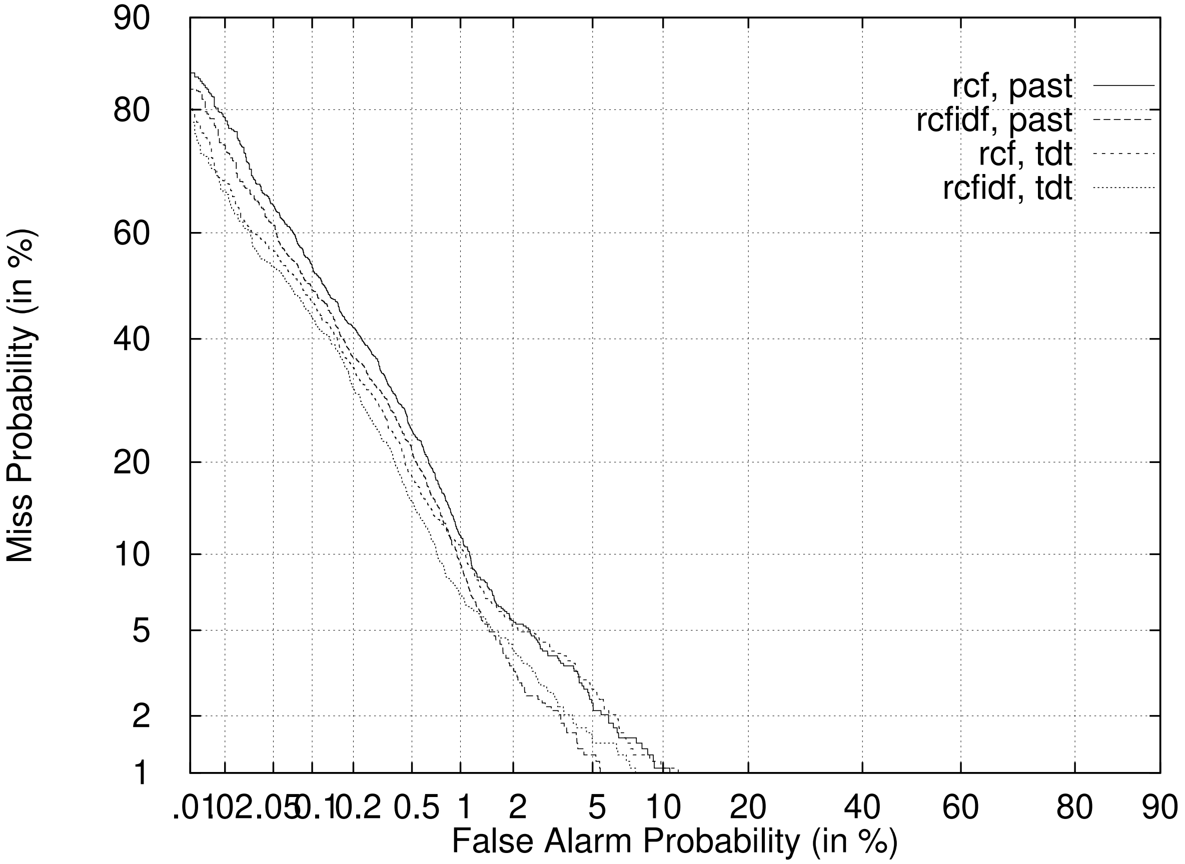


Figure 2: DET curve showing effects of training corpus on evaluation. For each of the TDT-only corpus and the larger corpus, two queries are run: one with the features weighted by rcf, the other with them weighted by rcf times incremental idf.

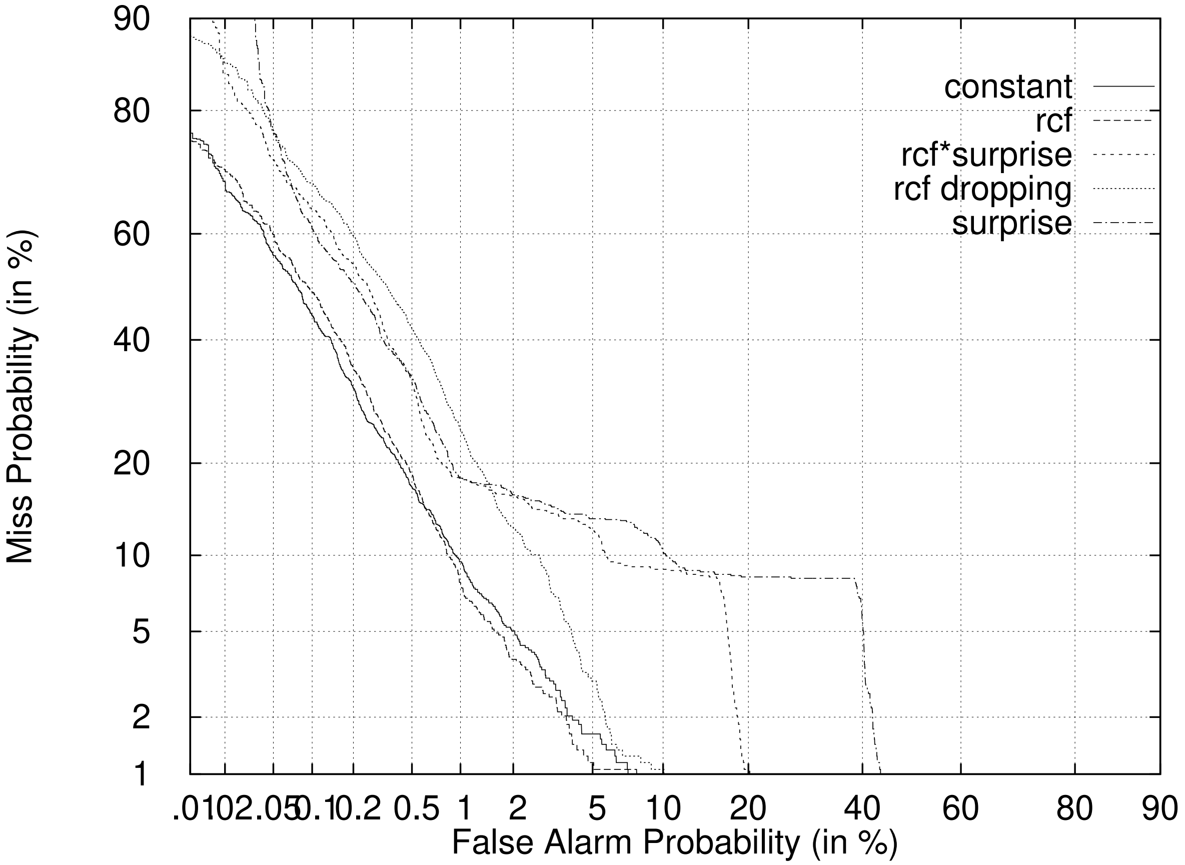


Figure 3: Various methods of weighting features, showing how the weights have widely ranging impacts on effectiveness.

the query features' weights is helpful, whereas using incremental idf information from the "past" corpus hurts in both cases when compared to using the less-accurate "tdt" idf statistics. We find this result surprising, and believe it is suggests that further investigation of the use of idf is warranted (we briefly explore one alternative, "surprise," below).

The rcf and rcfidf weighting approaches resulted in very similar DET curves. To demonstrate that this effect is not guaranteed, we ran four additional weighting schemes and plotted their DET curves in Figure 3. The two best-performing runs are rcf and using a constant value for the weights. Since a small number of terms appeared to be useful, we tried scaling the weights back rapidly after the first several features. That curve, "rcf dropping" is the one with a similar shape but shifted to the upper right. The worst performing curves are two that incorporate a notion of "surprise" (discussed below).

For comparison with a more familiar presentation of the quality of the runs, Figure 4 shows the same five runs on a recall/precision graph. The curves show a nearly identical preference among the systems, except that the "rcf dropping" run has very high precision at recall of 0.0 and shows the worst performance at high recall.

4.3 Surprising features

It is a characteristic of news reporting that stories about the same event often occur in clumps. This effect is particularly true for unexpected events (e.g., disasters or major crimes) where the news media exhibit strong interest in a story and report in nearly endless detail about it. As the triggering event fades into the past, the stories discussing the event similarly fade. For example, Figure 5 shows how many stories appeared per

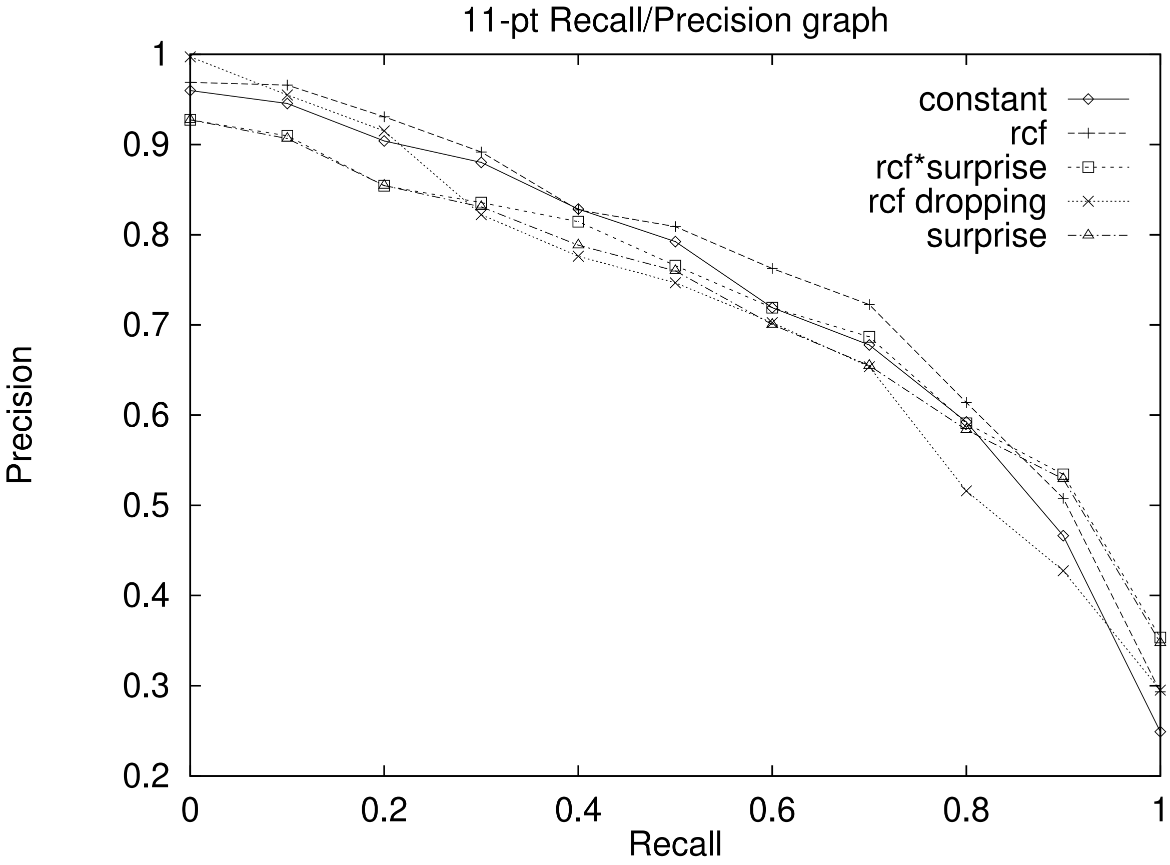


Figure 4: Recall/precision graph of the runs in Figure 3 for comparison.

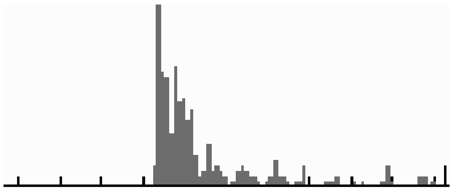


Figure 5: Picture of the OK city bombing reportage over time.

Feature	Surprise	rcf	rdf	Event
Kobe	1.29	19	4	Kobe
427	2.30	5	3	USAir 427 crash
cessna	1.09	5	4	Cessna
f-16	0.13	14	4	F-16
dna	0.11	15	4	OJ & DNA
lawn	0.05	17	4	Cessna
quake	0.13	13	3	Kobe
OKCity	0.25	12	4	OKCity
Breyer	0.22	36	4	Breyer
Intel	0.14	35	4	Pentium chip flaw
Rosario Ames	inf	14	3	Spy
bosnia	0.00	50	4	F-16
earthquake	0.04	27	4	Kobe
death	0.00	10	2	Salvi
death	0.00	5	3	OKCity

Table 1: Surprise values for a few words/phrases and a few events. The feature is shown along with its surprise value, the number of time it occurs in the $N_t = 4$ training set, the number of those 4 stories it occurs in, and the name of the event being considered.

day in the TDT corpus for the Oklahoma City bombing event. The sudden rise and then gradual fall of the stories is characteristic of this type of event.

A second characteristic of news coverage is that the people, places, and other items of interest in a story are likely not to have been mentioned very often in the past. This supposition is obviously not true for all features (e.g., the name of the President of the U.S. is likely to re-occur), but there must be *something* about the story that makes its appearance worthwhile. We call those features that have not occurred recently *surprising*.

An analysis of the events in this corpus shows that this effect is almost always true. We measure surprise based on the distance between this occurrence of a word and all past occurrences:

$$\mathit{Surprise(word,docid)} = \left(\sum_{i=1}^{df_{word}-1} \frac{1}{\log(docid - id_{wordi})} \right)^{-1}$$

where df_i is the number of stories to date containing word i and id_i is the sequence number of the most recent story that contained word i. The formula is the inverse of the sum of the inverses of the log of the distances from this word to all of its previous occurrences.[3] Table 1 shows the surprise values of selected words for some of the events. Of interest are words like *kobe* that are very surprising and occur frequently in the $N_t = 4$ training stories and *earthquake* which is entirely unsurprising but still common in those same stories. In general, we find that words that are common in the training set but that have little to no surprise value represent "topic" features covering broad news topics such as politics, death, destruction, and warfare. (We expect that topic- and event-level features can be combined in a meaningful way, perhaps with the topic features providing a form of "query zone."[18])

Two of the runs in Figure 3 represent the tracking ability of queries generated using surprise values. Because many of the "surprising" features appear to be strong indicators of the event being discussed, we had expected they could be used to build superior tracking queries. Unfortunately, the evaluation does not support that hope. Two problems arise: (1) the surprising words do not provide a broad enough coverage to capture all stories on the event (e.g., omitting "bosnia" for an event in that area of the world because it is not a surprising word), and (2) many of the words are useless for retrieval, either because they are misspellings

[3] This particular formula is primarily *ad hoc* to explore its value, though it is supported by some data exploration and empirical evidence. An information theoretic or probabilistic measure may prove more appropriate when we have a better understanding of the task.

or because they are surprising by chance (e.g., the name of someone interviewed). We have been unable to find a way to use surprising features as part of the query construction, though we continue to be optimistic that some method is possible. For retrospective tasks, where a feature's occurrence characteristics can be measured *after* its "surprising" appearance, we expect that a measure of a feature's surprise value and its subsequent longevity may provide more useful information.[4]

4.4 Other approaches

The TDT final report[1] includes results from the two research groups other than University of Massachusetts (UMass): Carnegie Mellon (CMU), and Dragon Systems. Each group reported results that were roughly comparable.

- CMU used a k-nearest neighbor classification for Event Tracking, comparing how close a new story was to the N_t training stories' vectors as compared to the vectors for the non-relevant stories in the training set.
- CMU also used a decision tree classification algorithm that selects features that distinguish the event from the non-relevant training stories. Decision trees were very successful for $N_t > 4$, though they have a disadvantage that they cannot easily output a confidence score that produces meaningful DET curves.
- Dragon leveraged their language model approach to speech recognition to track events. Given the N_t training stories, they constructed a small language model for the event. New stories were compared to that language model as well as a large number of background language models. If the story fits the event language model better than the background models, it is "tracked."

All of the approaches tried in the TDT pilot study were successful, although they were more complicated and slightly less effective than the simple techniques we have shown above. (The TDT workshop results were for the pilot study and do not necessarily represent the best results obtainable for the methods tried.)

5 Adaptive tracking

One of the reasons for a query's inability to track stories is that the discussion of an event changes over time. This effect is particularly well illustrated by the Oklahoma City bombing event. When the bomb exploded outside the Murrah building, its origin was a mystery. Six days later, Timothy McVeigh was arrested and charged with the crime. Indeed, there is no mention of McVeigh until the 61st story that is relevant to that event, so using the approaches of Section 4, it is impossible for his name to appear in a query for any value of N_t less than that. Several other events exhibit similar reporting characteristics, and a tracking method that accommodates the shifting reportage should prove useful. The issue is similar to drifting queries in information filtering.[2, 11]

We have implemented an adaptive version of the tracking system that can rebuild the query after it "tracks" a news story on a given event. This idea is a form of unsupervised learning and except for its incremental nature, is similar to the notion of "pseudo-relevance feedback" that has proved highly successful at TREC workshops.[16]

When a tracking query is first created from the N_t training stories, it is also given a threshold. We used an initial threshold of 0.8 for all events (recall that scores range from zero to just over one). During the tracking phase, if a story S scores over that threshold, we assume that S is relevant and the query is regenerated as if S were among the N_t training stories—so there are $N_t + 1$ then $N_t + 2$ and so on training stories.

The adaptive approach is highly successful at generating superior queries, particularly at low false alarm rates. Figure 6 shows adaptive tracking runs for $N_t = 4$ with 10-feature queries weighted by rcfidf. The figure shows the impact of using different threshold values for deciding whether or not to regenerate a query. The higher the threshold, the less likely a query is to be regenerated. The figure shows the non-adaptive

[4] Preliminary studies in the Retrospective Detection task of the TDT pilot study[1] support this intuition.

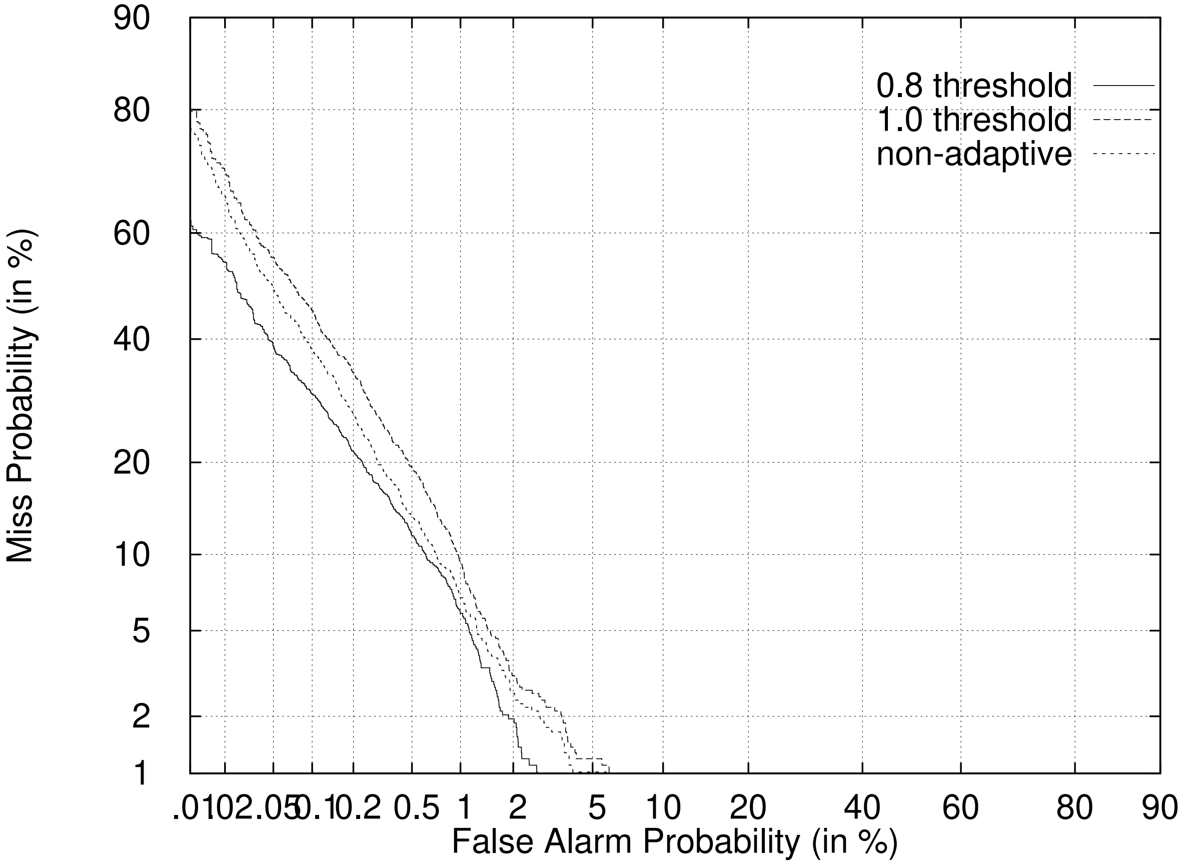


Figure 6: Varying thresholds for adaptive filtering.

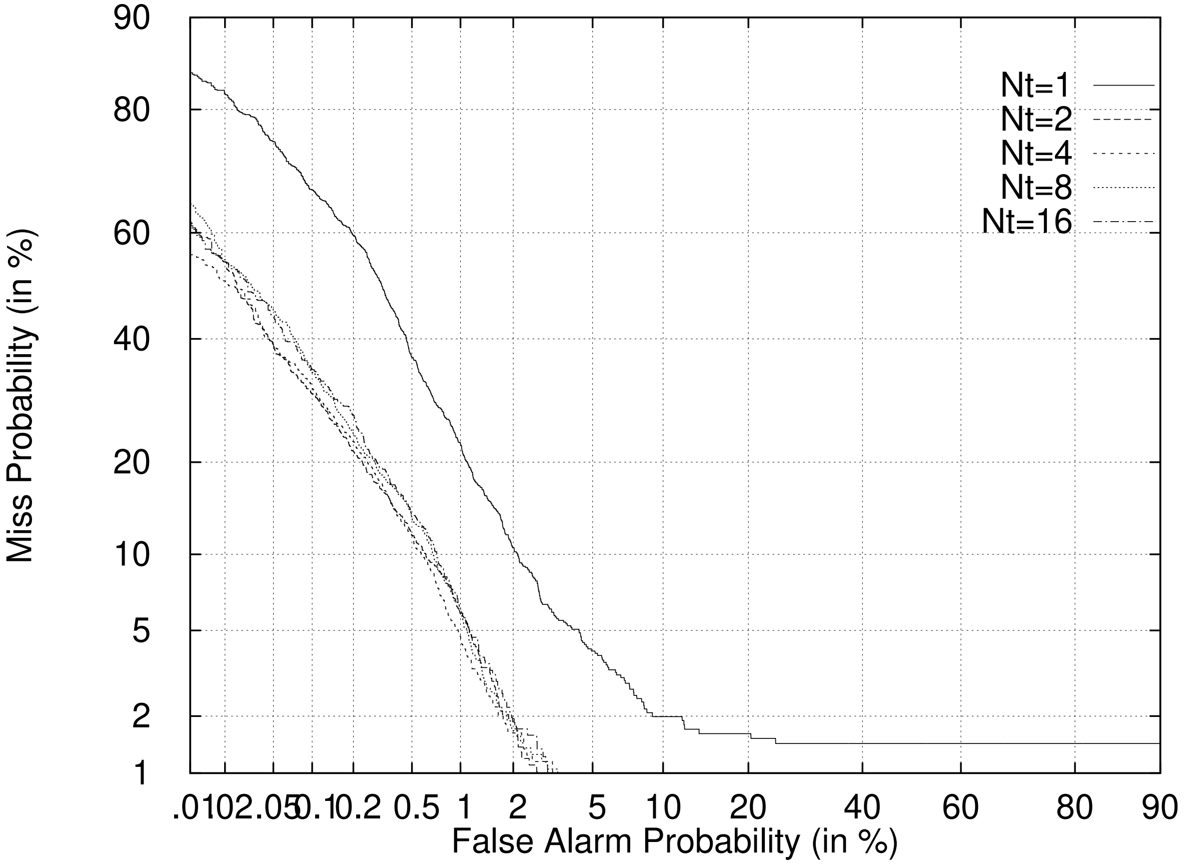


Figure 7: Adaptive filtering and the number of training instances.

run in the middle; a threshold of 0.8 improves performance, and one of 1.0 hurts it. Smaller thresholds (e.g., 0.6) cause performance to get consistently worse because they are adding stories that are less and less likely to be relevant.

One of the nice features of adaptive tracking is that as long as it works well, it allows the system to operate effectively with fewer sample stories. Figure 7 suggests that two sample stories is sufficient to achieve high-quality tracking: adding more causes almost no improvement—that effect as not achieved until $N_t = 4$ without adapting. Although the $N_t = 1$ curve is noticeably worse that then others, a comparison with Figure 1 shows that it still results in a substantial improvement in effectiveness.

We started off this section by talking about problems with words such as "McVeigh" and their not appearing in early stories. The final queries for $N_t = 4$ show that adaptive tracking successfully captures those features. [Timothy] *McVeigh* and [Terry] *Nichols* (the two suspects that have since been convicted) are both added to the Oklahoma City bombing event even though no mention is made of them until after the 60th stories; Scott O'Grady's name appears in the event describing the downing of F-16 pilot in Serbian territory, though it is six days and 38 stories later that the name is revealed.

We believe that adaptive event tracking is a necessary approach to this problem, as long as the system must work without user feedback after the N_t stories. We hypothesize that "surprise" information will be a useful indicator of valuable new features in adapting: a feature that appears suddenly and persists for a few stories is very likely to be important to the event.

6 Conclusions and future work

We have presented Event Tracking as a new variation on Information Filtering and a component of the Topic Detection and Tracking (TDT) initiative. Event Tracking is more narrowly defined than filtering, which allows for agreement in measures and area-specific processing for higher accuracy.

We also presented an evaluation methodology for Event Tracking that uses "rolling" training and test sets, and a Detection Error Tradeoff (DET) plot to measure accuracy on the system. We used miss and false alarm rates as our primary measures of effectiveness because the TDT tasks are seen as detection tasks. We suspect, however,that the results may be better understood if other measures are reported also, since they often capture different aspects of the task.

To illustrate the Event Tracking problem, we constructed and evaluated a system that built simple event queries. We showed that very few training stories are needed to build a high-quality query with a small number of features. We discussed the notion of surprising features and showed how adaptive tracking is a useful method for capturing those features in story sequences about disaster or crime events, and for reducing the number of training stories needed.

Significant advances in Event Tracking accuracy are most likely to be obtained using some limited form of story parsing and "understanding." It is likely to be useful to capture notions of who, what, where, when, why, and how—although the well-known past experience from IR suggests that the gains may not be large.

We have started investigating methods for recognizing names of people and places because they are key features in news reporting. Our hope is to find these features using primarily statistical methods (rather than natural language understanding) so that our work is more likely to extend to other areas. We will also take advantage of the shallow part of speech tagger that we have already used.

Acknowledgments

We thank Charles Wayne, George Doddington, Yiming Yang, Jaime Carbonell, and Jon Yamron with whom we worked to define the Topic Detection and Tracking tasks. George Doddington deserves particular credit for his guidance in establishing the evaluation methods for the Event Tracking task. A suggestion of Bruce Croft's about statistically anomalous features led eventually to the idea of "surprising" features. Finally, we appreciate Mike Scudder's help with the evaluation software.

The TDT initiative is a DARPA-sponsored project that supported this work. This material is also based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

References

[1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998. Forthcoming.

[2] James Allan. Incremental relevance feedback for information filtering. In *Proceedings of SIGIR '96*, pages 270–278, 1996.

[3] James Allan. Building hypertext using information retrieval. *Information Processing & Management*, 33(2):145–159, 1997.

[4] Chris Buckley, Gerard Salton, and James Allan. The effect of adding relevance information in a relevance feedback environment. In *Proceedings of SIGIR '94*, pages 292–300, 1994.

[5] Jamie Callan. Document filtering with inference networks. In *Proceedings of SIGIR '96*, pages 262–269, 1996.

[6] Christina Carrick and Carolyn Watters. Automatic association of news items. *Information Processing & Management*, 33(5):615–632, 1997.

[7] G. DeJong. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3:251–273, 1979.

[8] M. D. Dunlop. Time, relevance and interaction modelling for information retrieval. In *Proceedings of SIGIR '97*, pages 206–213, 1997.

[9] P.J. Hayes, L.E. Knecht, and M.J. Cellio. *A News Story Categorization System*, pages 518–526. Morgan Kaufmann Publishing, San Francisco, 1997.

[10] Karen Sparck Jones and Peter Willett, editors. *Readings in Information Retrieval.* Morgan Kaufmann Publishing, San Francisco, 1997. Chapter 4, pages 167-256.

[11] W. Lam, S. Mukhopadhyay, J. Mostafa, and M. Palakal. Detection of shifts in user interests for personalized information filtering. In *Proceedings of SIGIR '96*, pages 317–325, 1996.

[12] David D. Lewis. The TREC-5 filtering track. In E. M. Voorhees and D. K. Harman, editors, *The Fifth Text REtrieval Conference (TREC-5)*, pages 75–96, November 1997. NIST Special Publication 500-238.

[13] David D. Lewis and Kimberly A. Knowles. Threading electronic mail: A preliminary study. *Information Processing & Management*, 33(2):209–218, 1997.

[14] A. Martin, T. Kamm G. Doddington, M. Ordowski, and M. Przybocki. The det curve in assessment of detection task performance. In *Proceedings of EuroSpeech'97*, volume 4, pages 1895–1898, 1997.

[15] Brij Masland, Gordon Linoff, and David Waltz. Classifying news stories using memory based reasoning. In *Proceedings of SIGIR '92*, pages 59–65, 1992.

[16] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-2. In D. K. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 21–34, March 1994. NIST Special Publication 500-215.

[17] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983. Chapter 5, pages 157-198.

[18] Amit Singhal, Mandar Mitra, and Chris Buckley. Learning routing queries in a query zone. In *Proceedings of SIGIR '97*, pages 25–32, 1997.

[19] Jean Tague-Sutcliffe. Measuring the informativeness of a retrieval process. In *Proceedings of SIGIR '92*, pages 23–36, 1992.

[20] Ellen M. Voorhees and Donna Harman. Overview of the fifth text retrieval conference. In E. M. Voorhees and D. K. Harman, editors, *The Fifth Text REtrieval Conference (TREC-5)*, pages 1–28, November 1997. NIST Special Publication 500-238.

[21] Ellen M. Voorhees and Donna Harman. Overview of the sixth text retrieval conference. In E. M. Voorhees and D. K. Harman, editors, *The Sixth Text REtrieval Conference (TREC-6)*, 1998. NIST Special Publication, forthcoming.

[22] Jinxi Xu, John Broglio, and W. Bruce Croft. The design and implementation of a part of speech tagger for english. Technical Report IR-52, University of Massachusetts Center for Intelligent Information Retrieval, 1994.