Detections, Bounds, and Timelines: UMass and TDT-3

James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan

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

ABSTRACT

This report presents the system used by the University of Massachusetts for its participation in three of the five TDT tasks this year: detection, first story detection, and story link detection. For each task, we discuss the parameter setting approach that we used and the results of our system on the test data. In addition, we use TDT evaluation approaches to show that the tracking performance that sites are achieving is what is expected from Information Retrieval technology. We further show that any first story detection system based on a tracking approach is unlikely to be sufficiently accurate for most purposes. Finally, we present an overview of an automatic timeline generation system that we developed using TDT data.

1. BASIC SYSTEM

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 was originally developed for the 1999 summer workshop at Johns Hopkins University's Center for Language and Speech Processing.[1]

1.1. Detection algorithms

Our system supports two models of comparing a story to previously seen material: centroid (agglomerative clustering) and nearest neighbor comparison.

Centroid In this approach, we group the arriving documents into clusters. The clusters represent topics that were discussed in the news stream in the past. Each cluster is represented by a *centroid*, which is an average of the vector representatives of the stories in that cluster.

Incoming stories are compared to the centroid of every cluster, and the closest cluster is selected. If the similarity of the story to the closest cluster exceeds a threshold, we declare the story old and adjust the cluster centroid. If the similarity does not exceed the threshold, we declare the story new, and create a new singleton cluster with the story as its centroid.

k-nearest neighbor The second approach, k-NN, does not attempt to explicitly model a notion of a topic, and instead declares the story new if it is not like any story seen before.

Incoming stories are directly compared to all the stories we have seen before. The most similar k *neighbors* are found, and if the story's similarity to the neighbors exceeds a threshold, the story is declared old. Otherwise the story is declared new.

1.2. Similarity functions

One important issue in our approach is the problem of determining the right similarity function. We considered four functions: cosine, weighted sum, language models, and Kullbach-Leiblar divergence. The critical property of the similarity function is its ability to separate stories that discuss the same topic from stories that discuss different topics.

Cosine The cosine similarity is a classic measure used in Information Retrieval, and is consistent with a vector-space representation of stories. 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.

$$\left(\sum q_i d_i\right) / \sqrt{\left(\sum q_i^2\right)\left(\sum d_i^2\right)}$$

(Note that if $\vec{q}$ and $\vec{d}$ have unit length, the denominator is 1.0 and the angle is calculated by a simple dot product.) Cosine similarity tends to perform best at full dimensionality, as in the case of comparing two long stories. Performance degrades as one of the vectors becomes shorter. Because of the built-in length normalization, cosine similarity is less dependent on specific term weighting, and performs well when raw word counts are presented as weights.

Weighted sum The weighted sum is an operator used in the *InQuery* retrieval engine developed at the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts. *InQuery* is a Bayesian inference engine with transition matrices restricted to constant-space deterministic operators (e.g., AND, OR, SUM). Weighted sum represents a linear combination of evidence with weights representing confidences associated with various pieces of evidence:

$$\left(\sum q_i d_i\right) / \left(\sum q_i\right)$$

where q represents the *query* vector and d represents the *document* vector. For instance, in the centroid model, cluster centroids represent *query* vectors which are compared against incoming *document* vectors.

Weighted sum tends to perform best at lower dimensionality of the query vector q. In fact, it was devised specifically to provide an advantage with short user requests typical in IR. The performance degrades slightly as q grows. In addition, weighted sum performs considerably better when combined with traditional tf·idf weighting (discussed below).

Language model Language models furnish a probabilistic approach to computing similarity between a document and a topic (as in centroid clustering) or two documents (nearest neighbour). In this

approach, previously seen documents (or clusters) represent models of word usage, and we estimate which model M (if any) is the most likely source that could have generated the newly arrived document D. Specifically, we are estimating $P(D|M)/P(D)$, where $P(D)$ is estimated using the background model $P(D|GE)$ corresponding to word usage in General English.

By making an assumption of term independence (unigram model), we can rewrite $P(D|M) = \prod_i P(d_i|M)$, where d_i represent individual tokens in D. We use a maximum likelihood estimator for $P(d_i|M)$, which is simply the number of occurrences of d_i in M divided by the total number of tokens in M. Since our models may be sparse, some words in a given document D may have zero probability under any given model M, resulting in $P(D|M) = 0$. To alleviate this problem we use a smoother estimate $P(d_i|M) = \lambda P_{ml}(d_i|M) + (1-\lambda)P(d_i|GE)$, which allocates a non-zero probability mass to the terms that do not occur in M. We set λ to the Witten-Bell[6] estimate $N/(N+U)$ where N is the total number of tokens in the model and U is the number of unique tokens. (Note that since detection tasks are online tasks, we may enounter words not in GE, and so we smooth GE in a similar fashion using a uniform model for the unseen words.)

Kullbach-Leiblar divergence Instead of treating a document D as a sample that came from one of the models, we could view D as a distribution as well, and compute an information-theoretic measure of divergence between two distributions. One measure we have experimented with is the Kullbach-Leiblar divergence, $KL(D,M) = -\sum_i d_i \log(m_i/d_i)$, where d_i and m_i represent relative frequencies of word i in D and M respectively (both smoothed appropriately).

1.3. Feature weighting

Another important issue is weighting of individual features (words) that occur in the stories. The traditional weighting employed in most IR systems is a form of tf·idf weighting.

tf·idf The *tf* component of the weighting represents the degree to which the term describes the contents of a document. The *idf* component is intended to discount very common words in the collection (e.g. function words). Below is the particular tf·idf scheme used in the InQuery engine:

$$tfcomp = \frac{tf}{tf + 0.5 + 1.5\frac{len_d}{len_{avg}}}$$

$$idfcomp = \frac{\log(N/df)}{\log(N+1)}$$

The *tf-comp* component has a general form of $tf/(tf+K)$, where *tf* is the raw count of term occurrences in the document, and *K* influences the significance we attach to seeing consecutive occurrences of the term in a particular document. The functional form is strictly increasing and asymptotic to 1.0 as *tf* grows without bounds. The effect is that we assign a lot of significance to observing a single occurrence of a term, and less and less significance to consecutive occurrences. This is based on the observation that documents that contain an occurrence of a given word w are more likely to contain successive occurrences of w.

The parameter *K* influences how aggressively we discount successive occurrences, and in *InQuery* is set to be the document length over average document length in the collection. This means that shorter documents will have more aggressive discounting, while longer stories will not assign a lot of significance to a single occurrence of a term.

The *idf-comp* component is the logarithm of the inverse probability of the term in the collection, normalized to be between 0 and 1. *N* denotes the total number of documents in the collection, while *df* shows in how many of those documents the term occurs. This particular *idf* formulation arises naturally in the probabilistic derivation of document relevance under the assumption of binary occurrence and term independence.

tf This weighting scheme is simply the actual *tf* value used in the *tfcomp* formula above—i.e., the number of times the term occurs in the story. The intuition behind omitting the *idf* component is that feature selection at other points in the process will choose only medium- and high-idf features with good discrimination value. As a result, the *tf*-only weighting scheme is less likely to work at high dimensionality when low-idf features will appear and need to be downweighted.

idf This weighting scheme is simply the raw *tf* component times the *idf* component of the tf·idf scheme. This weighting method boosts the importance of multiple occurrences of a feature over that given in the tf·idf scheme.

2. FIRST STORY DETECTION

We tuned parameters and made our choice of detection algorithms by running FSD experiments using the January-June TDT2 corpus. The first four months served as a training corpus and the last two months as a development corpus. We also ran experiments on the entire six months of data. Detection algorithms (centroid, k nearest neighbor), similarity measures (cosine, weighted sum), weighting schemes (tf·idf, idf, tf), and thresholds were varied. Parameter selection was made on the basis of DET curves and the topic weighted $(C_{\text{fsd}})\text{norm}$ values.

2.1. Parameter setting

We set some initial parameters by using the entire six months of training data. By a small margin the best DET curve was the one generated using a 1-NN clustering algorithm, cosine similarity, dimensionality of 1000, and tf·idf weighting scheme. A run using all the same parameters and an idf weighting scheme came in a close second. A tf weighting scheme and again, all the same other parameters came in third. Alternative clustering algorithms, wsum, and lower dimensionality all proved less effective.

Previous experiments indicated that the optimal threshold was around 0.2 so we tried the three most promising weighting schemes and threshold values around 0.2, specifically, 0.18, 0.19, 0.2, 0.21, 0.22 and 0.24. The following table shows $(C_{\text{fsd}})\text{norm}$ values for different thresholds applied to the 1-NN, cosine comparison, tf·idf weighting, 1000-feature system:

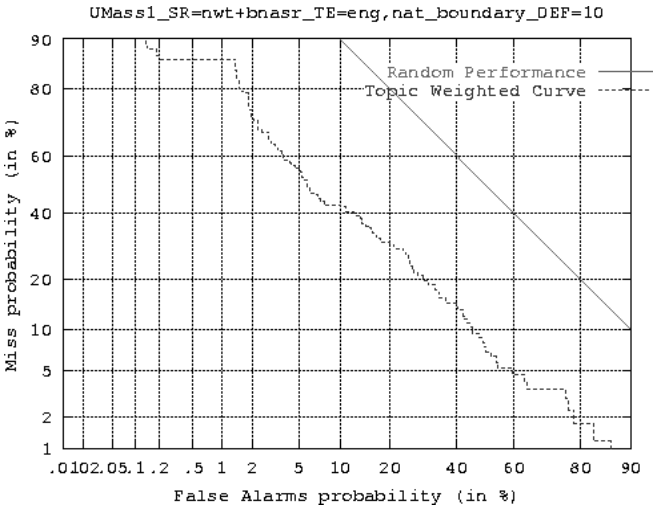
θ	(C_{fsd})norm (six months)
0.18	0.7596
0.19	0.7398
0.20	0.7024
0.21	**0.6504**
0.22	0.7315

To decide upon our actual parameter values, we used the four month-two month split of data. We varied the weighting scheme between tf·idf and idf, the two most promising weighting schemes, and again, used threshold values around 0.2. Once again, the top runs used a tf·idf weighting scheme. The optimal threshold for the training collection was again 0.21. The optimal threshold for development was 0.20. The following table shows some of the results:

Weights	θ	(C_{fsd})norm (Jan-Apr)	(C_{fsd})norm (May-Jun)
tf·idf	0.19	0.7112	0.6399
tf·idf	0.20	0.6412	**0.5959**
tf·idf	0.21	**0.5963**	0.6074
tf·idf	0.22	0.6415	–
idf	0.20	0.7158	–
idf	0.21	–	0.6156
idf	0.22	–	0.6081

On the basis of these results, our final runs used a 1-NN clustering algorithm, cosine similarity, dimensionality of 1000, the tf·idf weighting scheme, and a threshold of 0.21.

Our score on the primary first story detection evaluation (SR=nwt+bnasr TE=eng,nat boundary DEF=10) was 0.8110. The following graph shows the error tradeoffs:



3. STORY LINK DETECTION

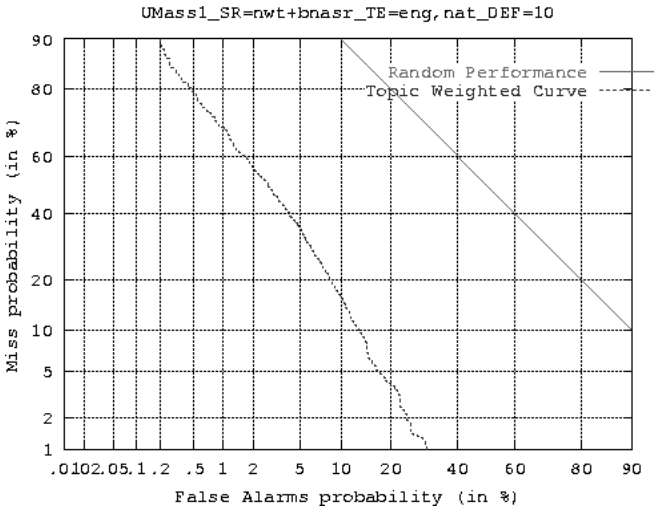
We made our choice of similarity measures, weighting schemes and thresholds by running story link experiments using the January-June TDT2 corpus. Similarity measures sampled were cosine, weighted sum, language model and Kullbach-Leiblar divergence. Weighting schemes sampled were tf·idf, idf, and tf. Previous experiments indicated that the best threshold was 0.1, so we tried values in that range. As usual, parameter selection was made on the basis of DET curves and the topic weighted (C_{link})norm values.

The following table shows the top 5 runs, all of which were cosine similarity, dimensionality of 1000, and idf weighting scheme. Within this set of parameters, threshold 0.08 yielded the lowest (C_{link})norm value.

θ	(C_{link})norm (six months)
0.06	0.1425
0.08	**0.1299**
0.10	0.1493
0.12	0.1741
0.14	0.2016

There was no easy way to divide this training collection into training and development collections so we did not verify these parameters by running them on four and two month subsets.

Our score on the primary link detection evaluation (SR=nwt+bnasr TE=eng,nat DEF=10) was 1.1385. The following DET plot shows the tradeoff between miss and false alarm errors:



Mid January, when NIST released a new link detection index file, we performed an additional run using all the same paramters. Our (C_{link})norm score for this run was 0.1427, almost an order of magnatude better than the orignial score. We do not yet know why link detection on this subset of stories was so much easier than the larger set.

4. DETECTION

As with first story detection, we ran wide parameter sweeps on the six month, January-June TDT2 corpus. We then confirmed our settings with narrower parameter sweeps and finer granularity using the four month and two month training and development corpora. We also checked our choice of parameters on different languages.

For languages, we tried eng-nat (English-only corpus in its natural language—i.e., English), mul-eng (English and Mandarin, with the Mandarin translated into English by SYSTRAN), and mul-nat (English and Mandarin, each in their own "natural" language).

Using the six month eng-nat collection we varied the clustering algorithm, weighting scheme, and threshold. The clustering algorithms sampled were 1-NN and centroid. The weighting schemes sampled were tf, idf and tf·idf. The thresholds were in the 0.20 to 0.30 range. We chose this threshold range on the basis of previous experiments. The optimal combination for the six-months of data was 1-NN, 1000 dimensionality, idf weighting, and threshold 0.20. The same trend showed up in the 4-month data. The following table shows some sample runs:

Weighting	θ	(C_{det})norm (six months)	(C_{det})norm (four months)
idf	0.20	**0.1806**	**0.1465**
	0.22	0.1902	0.1576
	0.26	0.1955	0.1705
	0.30	0.2175	–
tf	0.26	–	0.1938
	0.30	0.2246	0.1945

Further experiments on the four month eng-nat training corpus using 2-, 4-, and 8-NN clustering algorithms and a wider range of thresholds did not yield improvement.

Next we duplicated the English-only parameter sweep using the Multilingual collection with all stories in English (mul-eng). The results indicated that the parameter choice found using the English-only collection was, thus far, stable across languages. Both the four and six month collections confirmed the parameter choices made on the basis of the eng,nat collection. Once again, as the same runs in the following table show, we found the optimal parameter combination to be 1-NN, 1000 dimensionality, idf weighting, and threshold 0.2.

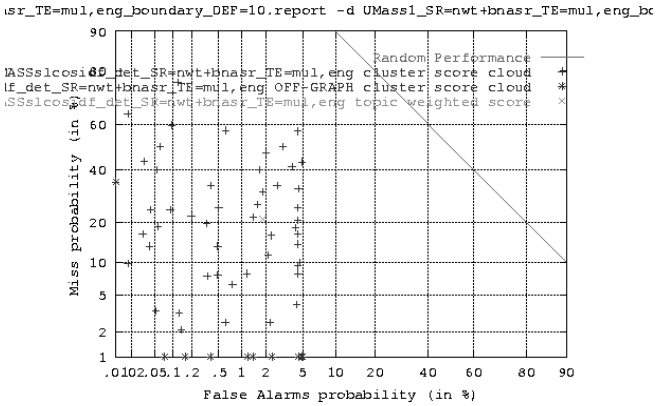
Weighting	θ	(C_{det})norm (six months)	(C_{det})norm (four months)
idf	0.20	**0.1901**	**0.1526**
	0.22	0.1970	0.1604
	0.26	0.2040	0.1796
	0.30	0.2314	0.2159
tf	0.30	0.2513	0.2239

Finally, we tried the same experiements using the Multilingual collection in the original languages (mul-nat) four and two month collections. The four month mul-nat training corpus again showed that the idf weighting scheme, 1-NN, and 0.2 threshold values were the most promising. This strengthened our confidence that this set of parameter choices is stable across languages. The following table shows some values for these runs:

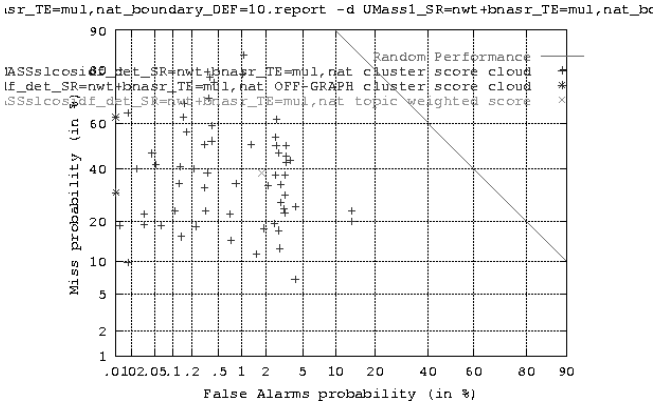
k for k-NN	Weighting	θ	(C_{det})norm (four months)
1	idf	0.16	0.1772
		0.18	0.1600
		0.20	**0.1549**
2	idf	0.18	0.2019
		0.20	0.1747

The 2-month mul,nat development corpus placed the optimal threshold at 0.21 but left all other parameters the same. On the basis of these results we used a 1-NN clustering algorithm, idf weighting scheme, and threshold of 0.2 for our official detection runs.

Our results on the primary detection evaluations were 0.3023 on the Multilingual in English task (SR=nwt+bnasr TE=mul,eng boundary DEF=10). The following DET curve shows the error tradeoffs for that run:



On the Multilingual in the original languages task (SR=nwt+bnasr TE=mul,nat boundary DEF=10) our result was 0.4682. The error tradeoff for that run was:



5. BOUNDS ON EFFECTIVENESS

In this section we show two things:

1. Tracking performance is approximately what we expect given state-of-the-art information filtering systems from TREC.
2. If an FSD system is built using a tracking system, it is extremely unlikely that FSD effectiveness can be satisfactory. We do not suggest that FSD is unsolvable, only that effective FSD is not a simple matter of improving tracking technology.

The work in this section is described in more detail elsewhere.[2, 1]

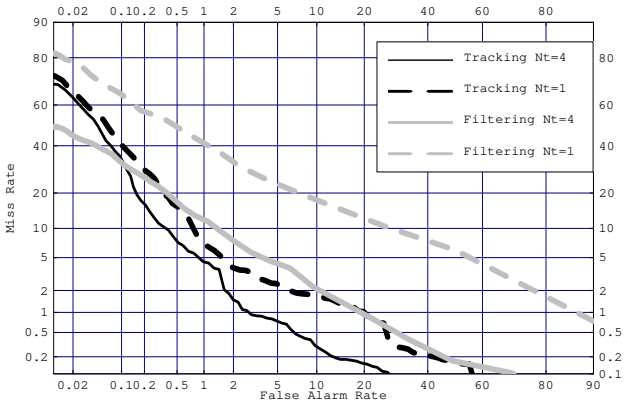


Figure 1: DET plot of two filtering and two tracking runs, each with the "query" generated from $N_t = 1$ or 4 stories.

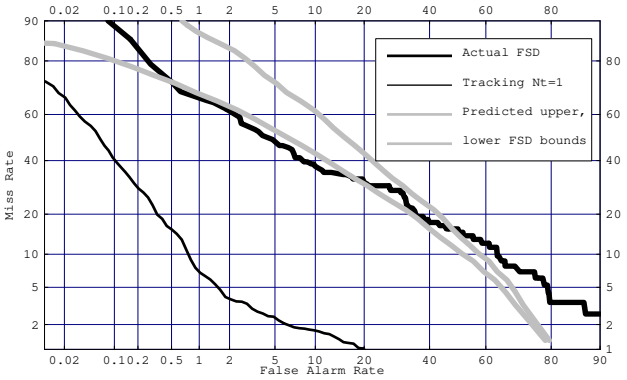


Figure 2: The lower-left graph is a tracking DET curve for $N_t = 1$. The upper part of the graph shows the lower- and upper-bound predicted performance for tracking-based FSD error rates in grey, as well as the actual system performance of an FSD system in black.

5.1. Expected TDT Performance

The DET curve of Figure 1 shows two tracking runs from the TDT-2 evaluation data. It also shows two runs from a TREC filtering task (modified to be more like tracking).

One thing that the graph shows is that tracking performance at $N_t = 4$ is near the performance that filtering achieves with similar starting information. Although the tasks were run on completely different corpora, and had different definitions, tracking performance is approximately what filtering performance predicts. We hypothesize that the wildly different performance of the tasks for $N_t = 1$ is because news topics are more focused (e.g., "Oklahoma City bombing") than TREC filtering queries (e.g., "drug legalization benefits"). As a result, a single story is a good representative of a news topic, but it might take several documents to isolate the information pertinent to a hidden query.

5.2. Bounds on FSD

One possible solution to FSD is to apply tracking technology. Intuitively, the system marks the first story of the corpus with a very high score (it *must* be the first story on any topic in the corpus). It then begins tracking that story. If the second story tracks, it is assigned a low FSD score. If it does not track (is not on the same topic as the first story), it is assigned a high FSD score, and the system starts tracking that one, too. At any point, the system is tracking numerous topics—in fact, if the system makes an FSD false alarm, it will be tracking some topics in multiple ways.

It should be clear that a perfect tracking system (for $N_t = 1$) yields a perfect FSD system. However, tracking systems are far from perfect. What sort of FSD performance can we expect from a state-of-the-art tracking system?

It is possible to derive expected FSD error rates from average TDT error rates (omitted here). The result will be lower- and upper-bounds on expected FSD performance. (We emphasize that the predictions only make sense if we assume that the FSD system uses an approach that is based upon tracking.) Figure 2 shows both the appropriate DET curves. Note that the FSD error rates fall nicely within the performance that is predicted by tracking. This result suggests that our FSD system is working about as well as we could expect.

5.3. Difficulty of improving FSD

The predicted and actual error rates of a tracking-based FSD system are in fact not very good: they are unacceptably high for all but a few applications, no matter what threshold on the DET curve is used.

We assume that "reasonable" FSD performance is approximately equal to the tracking DET curve shown in Figure 2 (the lower-left curve). A system that misses less than 10% of the first stories while generating only 0.5% false alarms is acceptable for many applica-

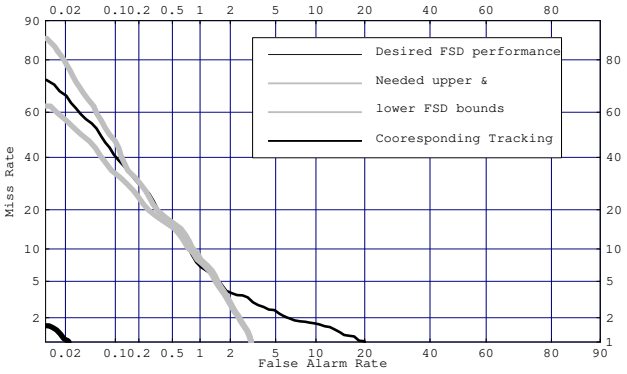


Figure 3: Shows desired FSD performance in black surrounded by reasonable confidence intervals. The extreme lower-left curve is the corresponding tracking performance.

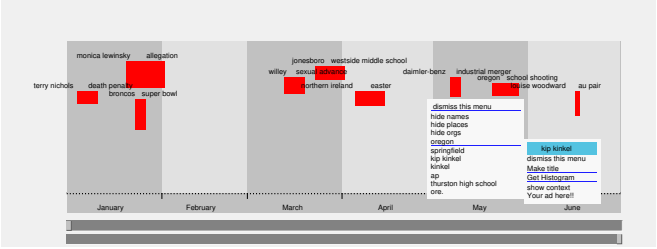


Figure 4: Overview of January - June, 1998. The topic labeled *monica lewinsky allegation* is the highest ranked topic, and the topic labeled *jonesboro westside middle school* is the second highest ranked. The pop-up on *oregon school shooting* shows significant named entities of *oregon*, *springfield*, *kip kinkel*, *kinkel*, *ap*, *thurston high school*, and *ore.* The other pop-up displays a submenu for obtaining more information on *kip kinkel*.

Figure 5: Detail, January 26 to March 13, 1998. Topics shown are *monica lewinsky allegation*, *broncos super bowl*, *karla faye tucker execution*, and *aviano cable car*. Additional phrases are displayed with the Karla Faye Tucker execution and the Aviano cable car crash.

tions.

Figure 3 shows the desired FSD curve (it is really just the tracking curve again) and lower- and upper-bounds on errors that encompass it. In order to achieve those bounds, we had to improve tracking performance for $N_t = 1$ by a factor of 20. The resulting DET curve is a small line segment in the lower left of the figure.

None of the research in TDT-1, TDT-2, and TDT-3 has resulted in a tracking DET curve that is substantially better than the ones in Figure 2. Further, as shown in Section 5.1, that level of effectiveness is comparable to that achieved by many years of filtering research at TREC. There is little reason to believe that tracking technology will ever improve 20-fold.

We have shown how to reduce the FSD problem to a tracking task. We have also shown that a given error rate in tracking results in substantially worse error rates in a corresponding FSD system. Most importantly, we have shown that there is little reason to believe that tracking-based FSD effectiveness can be raised to the point that the technology is widely useful.

6. AUTOMATIC TIMELINE GENERATION

We have developed a technique for determining the relative importance of the occurrence of extracted features within text. Our technique requires an explicitly time tagged corpus, such as TDT with its stories that arrive at known times. With our technique we are able to analyze extracted features (named entities and noun phrases) and explicitly rank how likely these features are to be high content bearing. We are then able to group these features into clusters that correspond strongly with the notion of "topic" as defined in the Topic Detection and Tracking (TDT) study. Figures 4 and 5 show examples of the system running. This work is described in more detail elsewhere.[5, 4]

With the model that tokens are emitted by random processes, we assume two hypotheses as defaults. The assumptions are 1: the random processes generating tokens are stationary, meaning that they do not vary over time, and 2: the random processes for any pair of tokens are independent. We use the χ^2 measure to look for features that violate those hypotheses. Details are omitted here.

We built a system that constructed timelines such as those shown. We were curious whether it was finding "reasonable" events, so we ran a small evaluation. We used the entire TDT-2 corpus for our experiment, training on the 4-month development set, and evaluating on the held-out 2 months. The corpus was tagged by BBN using the Nymble tagger[3], which identified 184,723 unique named entities. We also extracted noun phrases by running a shallow part of speech tagger[7], and labeling as a noun phrase any groups of words of length less than six which matched the regular expression (Noun|Adjective)*Noun. This led to a set of 1,188,907 unique noun phrases.

Our final run on the evaluation portion of TDT2 produced 146 clusters of those features (based on pairing features by χ^2 value and time, and imposing a threshold on which could be paired). We believe that the clusters of features found are indicative of the major news stories that were covered by the news organizations during the time spanned by the corpus. We felt that the clusters were highly suggestive of the major news stories and provided as good a summation as could be obtained by an unordered collection of features. To test this, we hired four students (three undergraduates and one graduate student) to evaluate the clusters.

Of the 146 clusters 79 were judged three times, and 67 had four judgments. The four evaluators found that the great majority of groups were indicative of a single topic (71.2%, 79.4%, 82.2% and 90.2% of the groups judged), and the pairwise overlap on the judgments of how many topics were contained in a group was 73.6%. However the overlap expected by chance was nearly 70%, and the pairwise Kappa statistics ranged from 0.045 to 0.315, with a (weighted) average value of 0.223. The Kappa statistic is a measure of inter-evaluator reliability, and a value of 0.0 indicates an overlap that would be expected by chance and a value of 1.0 indicates perfect overlap. A Kappa value of 0.233 indicates poor agreement among evaluators and that the data are not reliable. This can also be seen by looking at the scores given individual groups. Only twenty of the 146 were not judged to be a single topic by the majority of assessors, and of these twenty there were only three where the assessors unanimously agreed.

We also asked the assessors to compare the generated groups with the TDT2 topics and indicate if they agreed. Here the results were stronger. The (pairwise) overlap in topic/group matches was 86.7%, and the six pairwise Kappa statistics ranged from 0.600 to 0.785, with an average value of 0.699, indicating very good agreement. This indicates that if a topic is defined, the features our system selects are sufficient for recognizing the topic.

The groups of terms were automatically labeled and our assessors were asked to rate the usefulness of the label. Our assessors were asked to rank these on a six point Likert scale. In general our assessors felt that the labels were very poor, with an average rank of 2.8 ($1 =$ poor, $6 =$ excellent). Our assessors were in good agreement on the rankings, with the average standard deviation equaling 1.0.

We feel that the techniques presented in this study can make a significant contribution to the accessibility of information, as it allows the automatic generation of interactive overview timelines at modest cost. As archives of news, e-mails, historical newspapers, memos, and other such time based corpora become increasingly common in digital libraries we feel that this system, or one like it, will be a tremendous tool to allow broader access to electronic information.

7. CONCLUSION

The results that we have presented on the three detection tasks were acceptable, but not as high a quality as we would have liked. We believe that we have hit the limits of effectiveness that can be reached with simple IR-based approaches to story/topic comparison.

We spent considerable effort, including two months over the summer[1], working on FSD but were unable to achieve great improvements in the system. A major finding of that workshop, however, and one which we have extended since then[2], is the idea that tracking-based FSD systems cannot be effective enough. This result bolsters the idea that current approaches have hit their limits.

We believe that event-based information organization as realized in TDT requires substantially different approaches and ideas. We have briefly presented our work on automatic timeline generation, work that we believe serves as an example of moving TDT ideas in new directions. We hope that a richer set of ideas and directions will yield new approaches and techniques for addressing the existing TDT tasks, as well as new tasks that arise.

Acknowledgments

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 the National Science Foundation under grant number IRI-9619117, 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 authors and do not necessarily reflect the position or policy of the Government and no official endorsement should be inferred.

References

1. J. Allan, H. Jin, M. Rajman, C. Wayne, D. Gildea, V. Lavrenko, R. Hoberman, and D. Caputo. Topic-based novelty detection: 1999 summer workshop at CLSP, final report. Available at http://www.clsp.jhu.edu/ws99/tdt, 1999.
2. James Allan, Victor Lavrenko, and Hubert Jin. Comparing effectiveness in tdt and ir. Technical Report IR-197, University of Massachusetts, Department of Computer Science (CIIR), 2000. Conference submission.
3. D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*, pages 194–201. ACL, 1997.
4. Russell Swan and James Allan. Extracting significant time varying features from text. In *Eighth International Conference on Information Knowledge Management (CIKM'99)*, pages 38–45, Kansas City, Missouri, November 1999. ACM.
5. Russell Swan and James Allan. Automatic generation of overview timelines. Technical Report IR-198, University of Massachusetts, Department of Computer Science (CIIR), 2000. Conference submission.
6. I.H. Witten and T.C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37:1085–1094, 1991.
7. Jinxi Xu, J. Broglio, and W. B. Croft. The design and implementation of a part of speech tagger for english. Technical Report IR-52, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, 1994.