Dynamic Stopwording for Story Link Detection

Ralf D. Brown
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890 USA
ralf@cs.cmu.edu

ABSTRACT

Carnegie Mellon University entered two systems in the Story Link Detection track of the 2001 Topic Detection and Tracking (TDT) evaluation. These systems were one of our systems from the 1999 TDT evaluation [1], retuned for the new corpus, which had the third-best cost measure; and a new system that adds clustering and dynamically-generated stopwording, which had the best cost measure among all submissions for the default evaluation condition. This paper describes the enhancements which were made and some which were attempted but not used in the evaluation.

1. INTRODUCTION

With the ever-growing flood of information competing for a fixed amount of human attention, there is great interest in developing systems which can provide just the information a user is interested in from a real-time stream of data. The Topic Detection and Tracking project has been investigating various approaches to finding news stories which report on breaking events (topic detection, first-story detection) and those with continued coverage of events which the user has indicated are of interest (topic tracking). In 1999, an additional task was added to focus attention on the underlying primitive operation in all of the above – determining whether two news stories discuss the same thing (story link detection). Link detection is considered to be a primitive operation because if one can reliably determine linkages among stories, one can generate detection and tracking systems by checking whether incoming stories are linked to any previous stories. A story which has no links with any previous stories must be the first story about a new event, while a story which is linked to any of the stories which have already been tracked for an event should be tracked as well.

For the 2001 TDT evaluation, Carnegie Mellon University (CMU) entered two link-detection systems. One was a system from the 1999 TDT evaluation [1], retuned for the new corpus, and a new system that adds clustering and dynamically-generated per-cluster stopwording to the 1999 system. The motivation for the additions is that different events within a single class (such as different bombings or different airplane crashes) will use many of the same terms ("explosion", "crash"). If we can first determine a broad type of event for each story and then ignore the terms which stories in that class have in common, we should be better able to discriminate among similar events.

2. SYSTEM DESCRIPTION

The older system uses cosine similarity (after stopword removal) between TF*IDF-weighted term vectors to decide whether two news stories discuss the same subject. In addition to the actual words occurring in the story, terms are added to the vectors for words appearing within the first N words of each story to provide additional weight to the beginning of the story. The TF*IDF weights are updated each time a new source file (containing all stories from a news broadcast or half-hour sample of newswire) is added to the portion of the corpus which the link detector is permitted to examine. A dual threshold is used to determine whether the computed cosine similarity indicates a linkage between the two stories; one threshold is used when the two documents originate from the same type of source, and the other (typically lower) threshold is used for documents from different sources. For the 2001 evaluation, all Mandarin-language sources were grouped together, as were all English-language sources (i.e. a news story from Mandarin-language Voice of America would be treated as coming from the same source as a news story from Xinhua newswire).

The new system also uses cosine similarity, incrementally-updated TF*IDF weights, a global stopword list, and a dual threshold, but additionally performs clustering and cluster-specific stopwording. A single-pass incremental clustering method with configurable cluster-similarity metrics is used; for the TDT evaluation, the similarity metric selected was the root-mean-square of the cosine similarities between the new document and the members of each existing cluster. The resultant clusters have two uses in the link-detection system: selecting stopwords and applying a penalty when the two documents under test are in different clusters.

To select dynamic stopwords, the term frequencies for each individual cluster are computed. The N highest-frequency terms for each cluster which appear in no other cluster are made stopwords for that cluster. Since these terms appear only in the one cluster, they may actively harm recognition of a link to stories about the same event which happen to fall into a different cluster. And because they are common within the cluster, they reduce the differences between sto-

System YES/NO decisions

System	Miss	F.Alarm	Norm.Cost
1999	0.2381	0.0106	0.2899
2001	0.1778	0.0131	0.2421
change	-25.3%	+23.6%	-16.5%

Optimum According to DET Graph

System	Miss	F.Alarm	Norm.Cost
1999	0.1793	0.0175	0.2653
2001	0.1586	0.0159	0.2368
change	-11.5%	-9.1%	-10.7%

Table 1: System Performance Metrics

ries which are *not* linked.

Each test document then has the weights of the stopwords for the cluster containing it set to zero in its term vector prior to computing the cosine similarity, effectively removing the stopwords without actually altering the document stored in memory. After computing the similarity measure, a penalty is applied if the two test documents belong to different clusters.

Before using the highest-frequency cluster-unique terms, we implemented stopword selection by selecting for each cluster the terms with the highest information gain relative to that cluster. Unfortunately, this proved to be computationally intractable – even after several optimizations, a complete run would have taken an estimated 600 hours of CPU time, or very nearly the entire evaluation period.

A further enhancement which was disabled for the evaluation was a finite-state document rewriting engine for normalizing relative time expressions ("yesterday", "Tuesday", "last week", etc.). The normalization was expected to allow the system to better distinguish between different events using similar vocabulary, but proved to slightly worsen performance as measured by the TDT cost metric. There may simply not have been enough rewriting rules, as the existing rules are known not to handle all relative time expressions.

Based on experimental tuning, the parameters used for the evaluation were:

- prefer the first 100 words of the story
- 80 clusters
- 40 stopwords per cluster
- different-cluster penalty of 0.062

3. PERFORMANCE

The performance of the two systems in terms of missed links, false alarms, and the TDT cost metric is summarized in Table 1. This table shows both the cost computed from the systems actual YES/NO decisions and the cost computed using the threshold that minimized the cost while generating the DET curve.

Figure 1 shows the Decision-Error Tradeoff (DET) curve for the system with dynamic stopwording, while Figure 2 shows the performance without. Figure 3 compares the performances of all submitted systems for the default evaluation condition[1]. A DET curve is generated by applying a continuously-variable threshold to the scores output by the system, arbitrarily setting the decision to YES for all scores above the threshold and to NO for scores below the threshold, and computing miss and false alarm rates for each value of the threshold. The minimum TDT cost metric over all thresholds is also computed, and is displayed in the figures as the minimum DET cost (*Min DET* or *MDC*). This minimum cost is usually lower than the cost metric for the system's actual decisions and represents the best the system could perform under the assumption that its hard YES/NO decisions are made by thresholding on the reported score for the story pair. (If the system makes its YES/NO decisions from information other than the confidence score, the point representing the system's actual miss/false-alarm performance will lie off the DET curve.)

The discontinuity or "knee" in Figure 1 is the result of the binary nature of the different-cluster penalty. Since some linked stories end up in different clusters, the penalty is applied to these cases and thus the sliding threshold used to generate a DET curve does not reach those pairs until after a large number of non-linked stories have scores above the threshold. This raises a portion of the DET curve. It is clear that the penalty used in the evaluation was too large; brief experiments performed since the evaluation indicate that reducing the penalty from 0.062 to 0.03 will slightly lower the overall cost metric.

Although the dynamic stopwording improved the TDT cost metric from 0.2899 to 0.2421 (a 16.5% reduction), that improvement came at a huge cost in runtime – the time to process the evaluation test set increased from 36 minutes to 23 hours, a factor of 40. On the other hand, the more sophisticated system also may have more stable performance with varying data, as the minimum cost found by varying the threshold to generate the DET curve was 0.2368 (-2.2% compared to the system's hard YES/NO decisions) with dynamic stopwords and 0.2653 (-8.5%) without. Thus, if one could tune on the evaluation data to use the optimum parameters for the test data rather than the optimum parameters for the training data, the gap between the two systems would be smaller (only a 10.7% reduction).

4. FUTURE WORK

As of this writing, we are in the process of tuning a new system which uses cluster-specific term weighting rather than cluster-specific stop words. The idea is to adjust the impact of all terms rather than just zeroing out a limited number of terms. As yet, per-cluster TF*IDF weighting has not yet been able to match the performance achieved by the submitted system.

Because a section of the DET curve for simple cosine similarity without dynamic stopwords lies well below the curve for dynamic stopwording, it should be possible to combine the two systems, selecting the decision from one or the other depending on the range of scores. Previous work in combining multiple engines [2] has shown that such a combination yields a system with more stable performance on varying test data and sometimes even improves overall performance on the development data.

[1] All figures and performance measures are available at `ftp://jaguar.ncsl.nist.gov/tdt/tdt2001/eval/-tdt2001_official_results_20011029/index.htm`

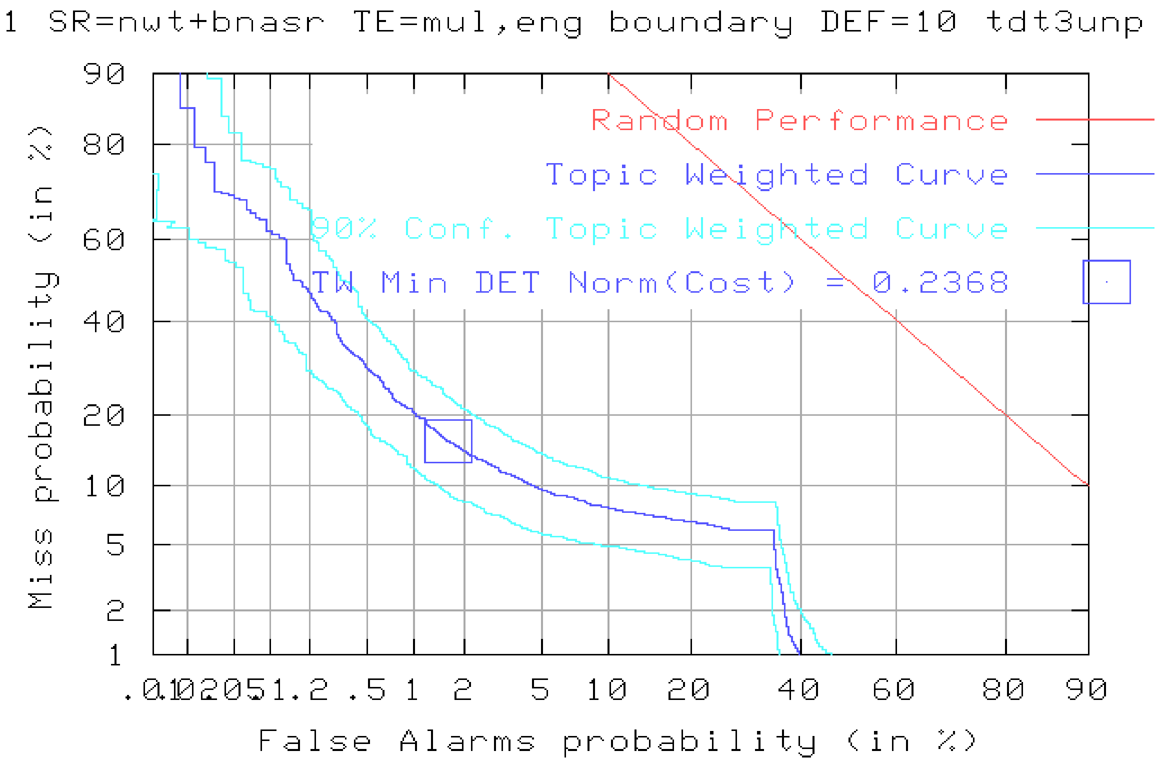

Figure 1: Performance with Dynamic Stopwording

5. REFERENCES

[1] R. D. Brown, T. Pierce, Y. Yang, and J. G. Carbonell. Link Detection – Results and Analysis, 2000. `http://www.nist.gov/speech/tests/tdt/tdt99/-papers/cmu_sld/`.

[2] Y. Yang, T. Ault, T. Pierce, and C. W. Lattimer. Improving Text Categorization Methods for Event Tracking. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000.

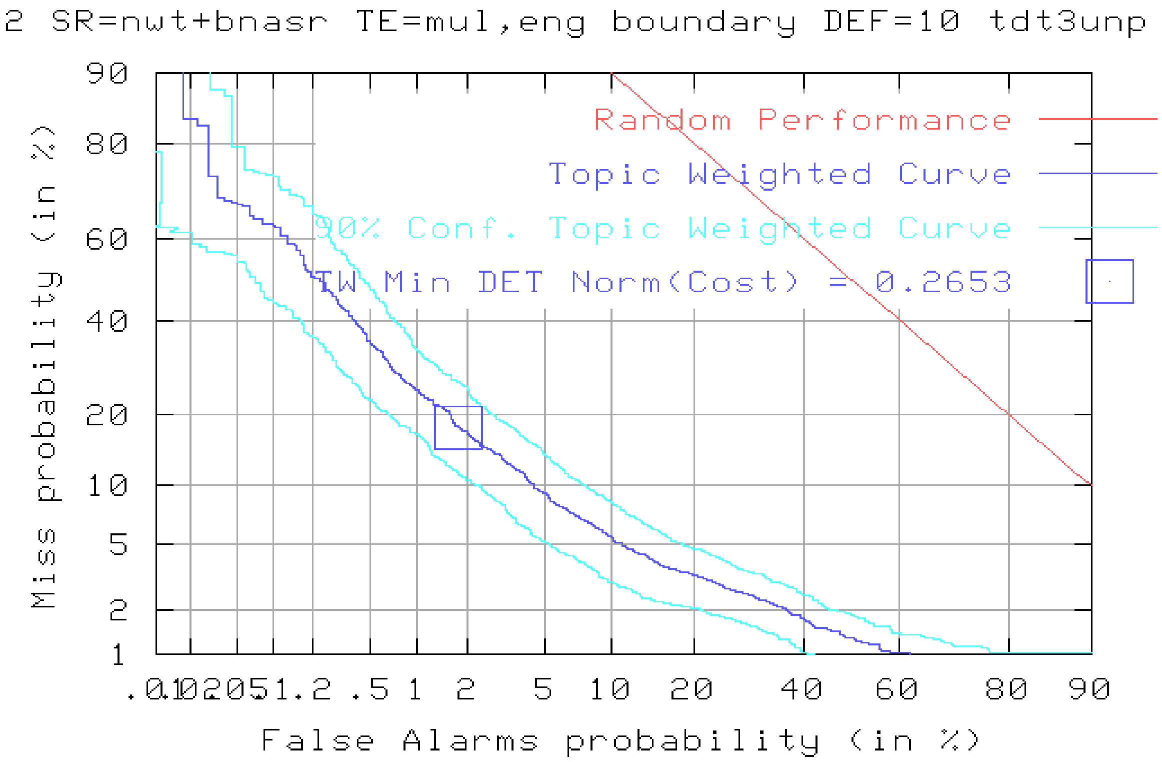

Figure 2: Performance without Dynamic Stopwording

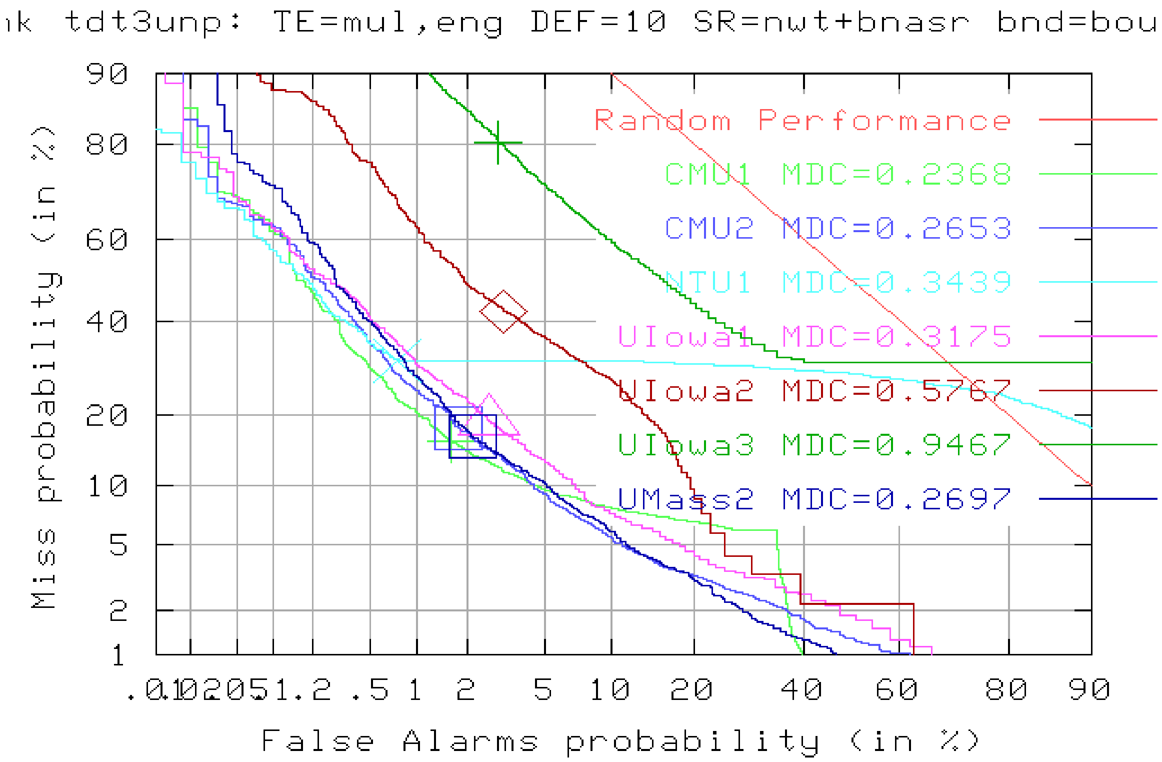

Figure 3: Cross-System Performance Comparison