

# Topic Detection in Broadcast News

Frederick Walls, Hubert Jin, Sreenivasa Sista, and Richard Schwartz

BBN Technologies  
70 Fawcett Street, Cambridge, MA. 02138, USA.  
hjin@bbn.com, schwartz@bbn.com

## ABSTRACT

We propose a system for the Topic Detection and Tracking (TDT) detection task concerned with the unsupervised grouping of news stories according to topic. We use an incremental  $k$ -means algorithm for clustering stories. For comparing stories, we utilize a probabilistic document similarity metric and a traditional vector-space metric. We note that the clustering algorithm requires two different types of metrics and adapt similarity metrics for each purpose. The system achieves a topic-weighted miss rate of 12% at a false accept rate of 0.22%.

## 1. Introduction

Topic Detection and Tracking (TDT) is a DARPA-sponsored initiative concerned with finding groups of stories on the same topic. It consists of three tasks: segmentation, tracking, and detection. We focus on the detection task, which is involved with the unsupervised grouping of stories that are on the same topic.

Story groupings are created through clustering, a technique that can be used to assign each story to one and only one group. In section 2, we propose a simple incremental  $k$ -means algorithm for clustering stories. The clustering algorithm requires a method for comparing stories with clusters. Therefore, section 3 details a probabilistic metric for this purpose. Section 4 describes methods for combining similarity metric scores into metrics useful for the two basic clustering tasks, selection & thresholding. Selection metrics find the most topical cluster to a story. Thresholding metrics provide a quantitative assessment of the topicality of a story. Section 5 presents the results of the experiments using the BBN detection system. A brief conclusion is given in section 6.

## 2. Clustering

We utilize an incremental  $k$ -means algorithm to cluster the data. We outline a basic incremental clustering algorithm. We then describe a technique to utilize the look-ahead granted by the TDT evaluation.

### 2.1. Incremental Clustering

One of the simplest clustering algorithms is the *incremental clustering* algorithm. This algorithm processes stories one at a time and sequentially, and for each story it executes a two-step process:

1. Selection: The most similar system cluster to the story is selected.
2. Thresholding: That story is compared to the cluster, and the system decides whether to merge the story with the cluster or to start a new cluster.

### 2.2. Incremental $k$ -means

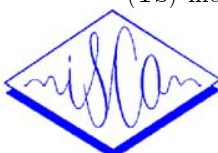
Although it is similar, the following algorithm is not precisely a  $k$ -means algorithm because the number of clusters  $k$  is not given beforehand. This algorithm involves iterating through the data that the system is permitted to modify and making appropriate changes during each iteration. More specifically:

1. Use the incremental clustering algorithm to process stories up to the end of the currently modifiable window.
2. Compare each story in the modifiable window with the old clusters to determine whether each should be merged with that cluster or used as a seed for a new cluster.
3. Modify all the clusters at once according to the new assignments.
4. Iterate steps (2)-(3) until the clustering does not change.
5. Look at the next few stories and goto (1).

This algorithm is able to restructure poor initial clusters but still process the corpus in a causal fashion with look-ahead. This algorithm also allows the number of clusters  $k$  to be a free parameter. The computational requirement is less imposing than the agglomerative clustering algorithm, especially for a larger look-ahead.

### 3. Probabilistic Similarity Metric

In addition to a conventional information retrieval (IR) vector-space approach, we utilize a probabilistic similarity metric called the BBN topic spotting metric. Probabilistic models offer a formal way of expressing computed quantities. A useful set of metrics for topic detection is the class of metrics that calculate  $P(C|S)$ . We shall analyze one particular example of such a metric, the BBN topic spotting (TS) metric [4].



The BBN TS metric is derived from Bayes’ Rule:

$$p(C|S) = \frac{p(C) \cdot p(S|C)}{p(S)}, \quad (1)$$

where  $p(C)$  is the *a priori* probability that any new story will be relevant to cluster  $C$ . If we assume that the story words  $s_n$  are conditionally independent, we get:

$$p(C|S) \approx p(C) \cdot \prod_n \frac{p(s_n|C)}{p(s_n)}, \quad (2)$$

where  $p(s_n|C)$  is the probability that a word in a story on the topic represented by cluster  $C$  would be  $s_n$ . We model  $p(s_n|C)$  with a two-state mixture model, where one state is a distribution of the words in all of the stories in the group, and the other state is a distribution from the whole corpus. That is, we have a generative model for the words in the new story.

The estimates for the general English state distribution and topic state distributions can be refined using the Expectation-Maximization (EM) algorithm [4]. This process allows new words to be added to the distributions and emphasizes topic-specific words. Therefore, the EM algorithm automatically assigns higher probabilities to words that are specific to the topic.

#### 4. Clustering Metrics

There are two types of metrics that are useful for the clustering algorithm we described: selection metrics and thresholding metrics.

##### 4.1. Selection Metric

A selection metric takes a story and outputs cluster scores such that the most similar cluster is found. Fortunately, we already found a metric that does this. The BBN topic spotting metric finds the most topical cluster to a story. This can be seen if we consider the problem as one of finding the most probable cluster given the story. More formally, In other words, from a set of clusters  $C_1, C_2, \dots, C_n$ , we attempt to find  $k$  such that:

$$k = \arg \max_i p(C_i|S). \quad (3)$$

#### Experimental Evidence

To test the effectiveness of the BBN topic spotting selection metric, we attempted a simple experiment. From each of the TDT-1, TDT-2 Jan-Feb, and TDT-2 Mar-Apr corpora (described in section ), a data set of human-generated clusters was extracted. Each cluster contained stories on one topic. Each story was removed from the data set one at a time and reclassified among the clusters in the data set. The story

was reclassified according to the highest-scoring cluster. If the highest-scoring cluster was not the cluster the story was drawn from, it was counted as an error. We report results using both the cosine distance and the BBN topic spotting (i.e., probabilistic) selection metrics.

The misclassification rates for each data set are given in table 1. The table indicates that the probabilistic selection metric reclassifies a larger percentage of stories correctly for all data sets. This suggests that the probabilistic metric is a more likely candidate for the selection problem than the cosine metric.

	TDT-1	TDT-2 (Jan-Feb)	TDT-2 (Mar-Apr)
Cosine dist.	1.32%	3.95%	0.18%
Probabilistic	0.09%	1.66%	0.00%

Table 1: Comparison of selection metrics according to misclassification rates for reclustered stories

##### 4.2. Thresholding Metric

The thresholding metric is discussed in the context of binary classification — given one story and one cluster, the story is either on the same topic as the cluster or not. The goal of a thresholding metric is to determine whether or not a story should be merged with a cluster. Such a metric is important for virtually any clustering algorithm one could conceive of, because it reveals whether or not a story belongs in a cluster. Therefore, we develop the following methods for combining scores and features from the system into an indicator about whether a story should or should not be merged with a cluster.

##### Score Normalization

One type of thresholding metric is the so-called “normalized score”, which is based on normalizing a single metric. To be effective, the normalization must minimize the effects of story and cluster size. The drawback of this approach is that the normalized score is only generated by one similarity metric.

Cosine distance metrics are naturally normalized — a score of 1 indicates that the stories are identical, and a score of 0 indicates the stories share no common words [3]. Therefore, a cosine distance metric could be used for thresholding. In particular, we use a cosine distance metric that smoothes the word counts and weights the vectors by an inverse document frequency (IDF) weight.

The BBN topic spotting metric is unfortunately not inherently well-normalized. The score varies with the size of the story compared. Fortunately, there are a few methods that can be used to normalize this metric.

For one normalization, we observe that the log probability produced by the topic spotting metric is proportional to the number of words in the story. Therefore, one possible normalization is to simply divide the log probability by the story length. While this produces a reasonable score, it is an *ad hoc* normalization.

Another normalization is to assume (by the Central Limit Theorem) that the log probabilities of a particular story  $S_i$  for different clusters are roughly distributed normally. This assumption can be justified if we view the individual word probabilities as independent random variables and assume that the story has a reasonably large number of words. Then, let  $\mu_i$  be an estimate of the mean of story log probabilities for cluster  $C$  and  $\sigma_i$  be an estimate of the standard deviation. Then, the normalized score for story  $S_i$  is given by

$$D(S_i, C) = \frac{\log p(S_i|C) - \mu_i}{\sigma_i}. \quad (4)$$

This normalization depends very little on the length of  $S_i$ , because any factor multiplying  $\log p(S_i|C)$  would cancel after the normalization. This normalized score is also a reasonable thresholding metric.

### Combining Normalized Scores

The official evaluation metric scores of various thresholding metrics are given in table 2. Combination 1 is a metric that decides that the story should be merged if the cosine distance or length-normalized topic spotting metrics are closer than a certain threshold. Combination 2 is the same as Combination 1, except that it uses the mean- and variance-normalized topic spotting metric rather than the length-normalized version.

These results indicate that the individual normalized similarity metrics produce good performance, especially on the topic-weighted score. Combining the metrics capitalizes on the strengths of each metric and produces improved scores. Therefore, our system utilizes the Combination 2 thresholding metric.

System	Value for $C_D$	
	Story-weighted	Topic-weighted
Cosine dist	0.0080	0.0025
Length-normed TS	0.0047	0.0031
Mean/sd-normed TS	0.0027	0.0014
Combination 1	0.0027	0.0022
Combination 2	0.0025	0.0013

Table 2: Comparison of different normalization schemes on TDT-2 Mar-Apr CCAP data

## 5. Results

We present some experimental results produced by the detection system.

### 5.1. Corpus and Evaluation

The Linguistic Data Consortium (LDC) has released a corpus for the 1998 TDT evaluation. The corpus, referred to as the TDT-2 corpus, consists of about 60,000 stories collected over a six-month period from both newswire and audio sources [1]. The TDT-2 corpus is subdivided into three two-month sets: a training set (Jan-Feb), a development test set (Mar-Apr), and an evaluation set (May-Jun). Because a detection system is not trained, there is little functional difference between the training set and development test set. Both sets can be used freely in the research and system design, but the evaluation set is withheld until the systems are evaluated.

LDC ([www ldc upenn edu](http://www ldc upenn edu)) annotated the corpus, and Dragon Systems ([www dragonsys com](http://www dragonsys com)) did automatic speech recognition for all the speech data. The average word error from Dragon’s automatic speech recognition was about 23%.

The official evaluation metric is a topic-weighted cost function [2], which compromises over miss errors and false alarm errors. In the 1998 TDT-2 evaluation,  $(0.02P(\text{miss}) + 0.98P(\text{fa}))$  was used to measure the performance of detection systems. Our 1998 BBN detection system, with a score of 0.0047, ranked the second best among all participants.

### 5.2. Evaluating the Parameters We Used

One concern with experiments conducted using the Jan-Feb and Mar-Apr data is the dependence of the thresholding metric decision thresholds on the corpus and human-chosen topics. We show in table 3 the dramatic difference between the thresholds chosen for the Jan-Feb data versus the Mar-Apr data. By tuning the metric thresholds, we can modestly improve the May-Jun set topic-weighted  $C_D$  to 0.0042 versus the evaluation result of 0.0047.

Topic-weighted results			
Data set	Cos thresh	TS thresh	$C_D$
Jan-Feb CCAP+NWT	-.95	-9.5	.0056
Mar-Apr CCAP+NWT	-1.0	-8.0	.0013
Mar-Apr ASR+NWT	-.85	-7.0	.0020
May-Jun ASR+NWT	-.95	-7.5	.0042

Table 3: Optimal clustering thresholds for different data sets

### 5.3. Differences Between Data Sets

Unfortunately, we find substantial differences between the different data sets that have been produced for TDT-2. Curiously, the Jan-Feb data has a few

topics that are very broad and a few that are very focused. This inconsistency is reflected in the system’s performance. The Mar-Apr data contains roughly 1/8 the number of labeled stories than the Jan-Feb data. Therefore, the Mar-Apr set contains smaller topics that are generally more consistent. Finally, the May-Jun set contains roughly 3 times the number of labeled stories as Mar-Apr. The May-Jun data set again has more variation, with several smaller topics and many larger topics. The scores are shown in table 4.

Topic-weighted results			
Data set	$P_M$	$P_{FA}$	$C_D$
CCAP+NWT Jan-Feb	0.1763	0.0021	0.0056
ASR+NWT Mar-Apr	0.0813	0.0004	0.0020
CCAP+NWT Mar-Apr	0.0435	0.0004	0.0013
ASR+NWT May-Jun	0.1292	0.0022	0.0047
CCAP+NWT May-Jun	0.1044	0.0023	0.0044

Table 4: Comparison of the same algorithm on different data sets (1 file look-ahead)

These results seem to suggest a correlation between the number of annotated stories and the cost function. The more stories that are labeled, the worse the system performs on the official evaluation metric. This effect is shown in table 5. The degradation in performance could be attributed to the lack of consistency in determining the human-annotated topics. The topics are determined separately for each data set by randomly sampling stories and heuristically determining the topic to which the sampled story belongs. Because the topics were determined months apart for each data set, the criteria used could be fundamentally different for each data set.

Data set	# of stories		Value of $C_D$	
	labeled	per topic	Story-wtd	Topic-wtd
Jan-Feb	3613	103.2	.0090	.0056
Mar-Apr	576	23.0	.0027	.0013
May-Jun	1312	38.6	.0035	.0044

Table 5: Results showing the correlation of  $C_D$  with average topic size (using CCAP+NWT data)

#### 5.4. Manual vs. Automatic Transcripts

The transcription method can have a significant effect on performance as well. ASR transcripts tend to have a very high error rate of about 23%, but the errors are relatively consistent. CCAP transcripts have a smaller error rate, but the errors are usually typographical errors and are often inconsistent. Even so, the combination of the newswire stories (NWT) with the CCAP data produces significantly better clusters than using newswire stories and ASR transcripts. These variations are illustrated in table 6.

Interestingly, in the tracking task, there is gener-

Topic-weighted results			
Data set	$P_M$	$P_{FA}$	$C_D$
ASR+NWT May-Jun	0.1292	0.0022	0.0047
CCAP+NWT May-Jun	0.1044	0.0023	0.0044
ASR+NWT Mar-Apr	0.0813	0.0004	0.0020
CCAP+NWT Mar-Apr	0.0435	0.0004	0.0013

Table 6: Comparison of ASR+NWT with CCAP+NWT (1 file look-ahead)

ally less degradation from using the ASR text versus CCAP text. This can be attributed to the training data that tracking systems are allowed combined with the consistency of the ASR errors. For example, a story that talks about “Iraq” might contain many consistent references to “a rock”, because the two are essentially homonyms. A detection system might split such a cluster into stories about Iraq and stories about rocks.

## 6. Conclusion

We discussed our system for clustering news stories according to topic. We utilized an incremental  $k$ -means clustering algorithm to group the stories. The clustering algorithm required two types of clustering metrics: selection and thresholding. For the selection metric, we used the BBN topic spotting metric. For the thresholding problem, we utilized a hybrid of the BBN topic spotting metric with a more conventional cosine distance metric. Finally, we presented some comparative results generated by our system.

## ACKNOWLEDGMENTS

This work was sponsored by the Defense Advanced Research Projects Agency and monitored by the Space and Naval Warfare Systems Command under Contract No. N66001-97-D-8501. The views and findings 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. Linguistic Data Consortium. Linguistic Data Consortium (LDC) TDT web site. <http://www ldc.upenn.edu/TDT/>.
2. National Institute of Standards and Technology (NIST). *The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan Version 3.7*, September 1998.
3. G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Massachusetts, 1989.
4. R. Schwartz, T. Imai, L. Nguyen, and J. Makhoul, “A Maximum Likelihood Model for Topic Classification of Broadcast News,” In *Proc. Eurospeech*, Rhodes, Greece, September 1997.