A System for New Event Detection

Thorsten Brants
thorsten@brants.net

Francine Chen
fchen@parc.com

Ayman Farahat
farahat@parc.com

Palo Alto Research Center
3333 Coyote Hill Rd, Palo Alto,
CA 94304

ABSTRACT

We present a new method and system for performing the New Event Detection task, i.e., in one or multiple streams of news stories, all stories on a previously unseen (new) event are marked. The method is based on an incremental TF-IDF model. Our extensions include: generation of source-specific models, similarity score normalization based on document-specific averages, similarity score normalization based on source-pair specific averages, term reweighting based on inverse event frequencies, and segmentation of the documents. We also report on extensions that did not improve results. The system performs very well on TDT3 and TDT4 test data and scored second in the TDT-2002 evaluation.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval; I.2.7 [**Computing Methodologies**]: Natural Language Processing

General Terms

Algorithms

Keywords

New Event Detection

1. INTRODUCTION

New Event Detection is the task of detecting stories about previously unseen events in a stream of news stories. A possible application for this task is to alert a news analyst when a new event occurs, e.g., an airplane crash, an earthquake, governmental elections, etc. The system avoids redundancy by not reporting the same event twice. Figure 1 presents the main idea of a New Event Detection system. A stream (or multiple streams) of news stories contains stories about different events. Stories on two different events are marked in the example, one with circles, the other one with diamonds. The first story on each event is to be flagged by the system. Apart from a stand-alone application that informs an analyst, New Event Detection can also be used in document summarization and document retrieval systems.

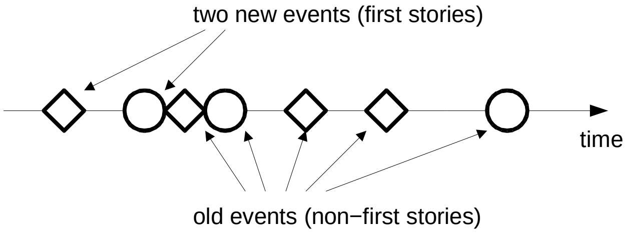
two new events (first stories)
time
old events (non−first stories)

Figure 1: New Event Detection in a stream of news stories. Two different events are marked by diamonds and cirlces. The first story on each event is to be flagged.

New Event Detection is one of the five tasks in the "Topic Detection and Tracking" evaluations performed each year by the National Institute of Standards and Technology.[1] New Event Detection is especially useful for the task of scanning multiple news sources for the latest news. It can be used in a categorization system to identify new categories of news stories and these stories can be used as examples of new categories. People who need to know the latest news when it happens, such as government analysts or financial analysts and stock market traders, can use New Event Detection to more quickly identify new events.

The structure of the paper is as follows. Section 2 gives a short overview of the best current approaches. Section 3 presents the basic model for new event detection that most current systems use. Section 4 introduces our new extensions and improved model. Section 5 presents experimental results. Section 6 lists extensions that did not improve results, and Section 7 gives conclusions.

2. RELATED WORK

Currently, TFIDF is the prevailing technique for document representation and term weighting for the New Event Detection task. All four systems evaluated in TDT-2002 (including ours) are using it. There are, however, differences

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'03, July 28–August 1, 2003, Toronto, Canada.
Copyright 2003 ACM 1-58113-646-3/03/0007 ...$5.00.

[1] The name of this task was changed to "New Event Detection" in 2002. This task was previously known as "First Story Detection".

in the application of TFIDF.

CMU uses it in combination with different clustering techniques for historical and lookahead data [10]. They first classify a new document into a number of broad topics, and then perform topic-conditioned novelty detection within each topic [11, 12]. The topic models include topic-specific stopword removal and topic-sensitive weighting of named-entities and other features.

IBM bases their decision on a single-link TFIDF measure plus n-gram overlap and temporal features. Except for a set of slides at the TDT-2001 meeting [6], no information about their system is available.

UMass also uses TDIDF and single link; all their TFIDF statistics are generated incrementally. Part of their model is described in [2]. Additionally, they use topic models and relevance models [8].

3. THE BASIC MODEL

This section presents the basic New Event Detection model that is similar to what most current systems use. We use it as a base for our extensions.

3.1 Pre-Processing

For pre-processing, we tokenize the data, recognize abbreviations, normalize abbreviations, remove stop-words, replace spelled-out numbers by digits, add part-of-speech tags, replace the tokens by their stems, and then generate term-frequency vectors.

3.2 Incremental TF-IDF Model

Our similarity calcuations of documents are based on an incremental TF-IDF model. In a TF-IDF model, the frequency of a term in a document (TF) is weighted by the inverse document frequency (IDF). In the incremental model, document frequencies[2] $df(w)$ are not static but change in time steps t. At time t, a new set of test documents C_t is added to the model by updating the frequencies

$$df_t(w) = df_{t-1}(w) + df_{C_t}(w) \tag{1}$$

where df_{C_t} denote the document frequencies in the newly added set of documents C_t. The initial document frequencies $df_0(w)$ are generated from a (possibly empty) training set. In a static TF-IDF model, new words (i.e., those words, that did not occur in the training set) are either ignored in further computations, or they are treated as having some small constant as their document frequency, e.g., $df = 1$. In the first case, new terms receive no weight at all, in the latter case, new terms receive too much weight. An incremental TF-IDF model uses the new vocabulary and assigns weights in accordance with their usage in new documents. This is an advantage because new events often introduce new vocabulary, and finding good weights for them should improve the model.

Very low frequency terms w tend to be uninformative. We therefore set a threshold θ_d, and only terms having

$$df_t(w) \geq \theta_d \tag{2}$$

are used at time t. Unless otherwise noted, we use[3] $\theta_d = 2$.

3.3 Term Weighting

The document frequencies as described in the previous section are used to calculate weights for the terms w in the documents d. At time t, we use

$$weight_t(d,w) = \frac{1}{Z_t(d)} f(d,w) \cdot \log \frac{N_t}{df_t(w)} \tag{3}$$

where N_t is the total number of documents at time t. $Z_t(d)$ is a normalization value with

$$Z_t(d) = \sum_w f(d,w) \cdot \log \frac{N_t}{df_t(w)} \tag{4}$$

if we use Hellinger distance, Kullback-Leibler divergence, or Clarity-based distance (see below), or

$$Z_t(d) = \sqrt{\sum_w \left[f(d,w) \cdot \log \frac{N_t}{df_t(w)} \right]^2} \tag{5}$$

if we use cosine distance.

3.4 Similarity Calculation

The vectors consisting of normalized term weights $weight_t$ are used to calculate the similarity between two documents d and q. In our current implementation, we either use the cosine distance

$$sim_t^c(d,q) = \sum_w weight_t(d,w) \cdot weight_t(q,w) \tag{6}$$

or Hellinger distance

$$sim_t^h(d,q) = \sum_w \sqrt{weight_t(d,w) \cdot weight_t(q,w)}. \tag{7}$$

Other possible similarity metrics include the Kullback-Leibler divergence, the symmetric form of it, Jensen-Shannon distance, or Clarity-based distance [5, 8], which have been found useful in other work [3].

3.5 Making a Decision

In order to decide whether a new document q that is added to the collection at time t describes a new event, it is individually compared to all previous documents d. We identify the document d^* with highest similarity to q:

$$d^* = \operatorname*{argmax}_d sim_t(q,d). \tag{8}$$

The value

$$score(q) = 1 - sim_t(q, d^*) \tag{9}$$

is used to determine whether a document q is about a new event and at the same time is an indication of the confidence in our decision. If the score exceeds a threshold θ_s, then there is no sufficiently similar previous document, thus q describes a new event (decision `YES`). If the score is smaller than θ_s, then d^* is sufficiently similar, thus q describes an old event (decision `NO`). The threshold θ_s can be determined by using labeled training data and calculating similarity scores for document pairs on the same event and on different events.

[2] The document frequency $df(w)$ denotes the number of documents in the collection that contain word w.

[3] Thresholds larger than 2 cannot be applied with empty or very small training sets. However, for the TDT task, an update is performed on groups of documents, where the size of the group varies, but is on the order of 20-30 or so. For this task, $\theta_d = 2$ works well with both small and large amounts of data.

Individual comparison to each document in the history has been shown to be superior to a clustering approach where new documents are compared to the generated clusters [10, 1].

3.6 Elimination of Short Documents

All TDT sources contain a number of very short documents that do not describe an event but are announcements, teasers, or other non-topical documents. Exclusion of very short stories from similarity calculations tends to improve results. An explanation for this is that teasers often mention different events, but according to the TDT labeling instructions they are not considered on-topic.

4. IMPROVEMENTS OF THE BASIC MODEL

In the following, we describe our new extensions and improvements of the basic model. These improvements were motivated by noting that the stories vary in a number of ways, including source, topic, and "rules of interpretation", and that differences in average similarity may result. We developed methods for normalizing the similarity scores based on these conditions. We also noted that stories about the same event often have parts in common, but each story may also have unique information. Our last extension makes use of this information to modify our basic model.

4.1 Source-Specific TF-IDF Model

Documents in the stream of news stories may stem from different sources. As an example, there are 20 different sources in the data for TDT 2002. Among them are ABC News, Associated Press, New York Times, Voice of America, etc. Each of the sources might have somewhat different vocabulary usage. For example, the names of the sources, names of particular shows, or names of news anchors are much more frequent in their own source than in the other ones. Additionally, there are more subtle differences in the preferred vocabulary. In order to reflect the source specific differences, we do not build one incremental TF-IDF model, but as many as we have different sources and use document frequencies

$$df_{s,t}(w) \tag{10}$$

for source s at time t. The frequencies are updated according to equation (1), but only using those documents in C_t that are from the same source s. As a consequence, a term like "CNN" receives a high document frequency (thus low weight) in the model for the source CNN and a low document frequency (thus high weight) in the model for the New York Times.

Instead of the overall document frequencies $df_t(w)$, we now use the source specific $df_{s,t}(w)$ when calculating the term weights in equation (3).

Sources s for which no training data is available (i.e., no data to generate $df_{s,0}(w)$ is available) might be initialized in two different ways:

1. Start with an empty model, i.e., $df_{s,0}(w) = 0$ for all w;

2. Identify one or more other but similar sources s' for which training data is available and use:

$$df_{s,0}(w) = \sum_{s'} df_{s',0}(w). \tag{11}$$

For the work presented here, we identified similar sources. This was done by creating a source hierarchy. When no training data is available for a particular source, we combine sources higher up in the hierarchy. As an example, we combine all English speech-recognized data in order to initialize an unknown English speech-recognized source.

4.2 Document Similarity Normalization

Some documents are similar to a large variety of documents (e.g., because they are on a very broad topic), while others are very distinctive. A high similarity of a broad topic document to some other document generally does not mean the same as a high similarity of a narrow topic document to some other document. In order to capture this difference, we calculate the average similarity $\overline{sim}(q)$ of the current document q to all previous documents in the collection. Instead of the original similarity $sim(q, d)$ between the new document q and an old document d, we use the adjusted similarity

$$sim'(q, d) = sim(q, d) - \overline{sim}(q) \tag{12}$$

Other normalizations could have been used, such as the ratio of the computed and the average similarity. However, such a normalization emphasizes deviations from the average much more than the linear difference. We observed that a linear normalization model is a better model of relative similarity.

4.3 Source-Pair Specific On-Topic Similarity Normalization

Documents that stem from the same source and that describe the same event tend to have a higher similarity than documents that stem from different sources and also describe the same event because of vocabulary conventions the sources adhere to. Similarly, one can argue that documents on the same event stemming from sources A and B have a different average similarity than documents from sources A and C describing the same event. In other words: if we have documents a, b, and c from sources A, B, and C, respectively, these have different expected similarity values:

$$E[sim(a, b)] \neq E[sim(a, c)] \tag{13}$$

In the special case of two stories a and a' on the same event and from the same source A we find

$$E[sim(a, a')] > E[sim(a, b)] \tag{14}$$

with b also on the same event but from a different source $B \neq A$. In order to reflect these differences, we adjust the similarity $sim'(q, d)$ as given in the previous section by the average similarity $E_{s(q),s(d)}$ of stories on the same event from the particular source pair that q and d are drawn from:

$$sim''(q, d) = sim'(q, d) - E_{s(q),s(d)} \tag{15}$$

where $s(q)$ and $s(d)$ denote the sources of documents q and d.

4.4 Using Inverse Event Frequencies of Terms

Events for the TDT tasks are further categorized into "rules of interpretation" (ROI). The ROI can be seen as a higher-level categoriztion of the events. As an example, the data contain the ROI *Elections* with two of its events being *Brazilian presidential elections* and *U.S. mid-term elections.* Stories on events from the same ROI can be expected to share some vocabulary (e.g., general election vocabulary). Because of this vocabulary overlap, we expect that events from the same ROI are more easily confused than events from different ROIs. Yang et al. [11] have explored this idea using topic-specific stopword removal, weighted use of named entities, and topic-sensitive feature weighting.

Terms that characterize a particular ROI but that do not distinguish particular events (e.g., the term *election*) should receive lower weight than terms that are highly informative about the event (e.g., the term *Cardoso*, the name of the former Brazilian president). However, the weight cannot be made too low in order not to confuse events that are from different ROIs but have an overlap in their event-specific vocabulary.

In order to address this problem, we will use event frequencies $ef(w)$ in addition to term document frequencies $df_{s,t}(w)$ for calculating term weights. The event frequency $ef(w)$ is defined as follows:

$$ef(w) = \max_{r \in ROI} ef(r, w) \tag{16}$$

where $ef(r, w)$ is the number of events that belong to ROI r and that contain term w. If appropriate information is available, the event frequencies can be adapted with time t. With a large enough training set they can also be generated specific to the sources of the processed documents. However, for the TDT tasks, the test data does not contain ROI labels and the amount of training data is small. Therefore, these counts will be generated from a fixed training set, independent of the source, and they do not change over time.

The best way of using the event frequencies is unknown yet. We adopted an ad-hoc technique. The original term weights are adjusted in the following way (cf. equation 3):

$$weight'_t(d, w) = \frac{1}{Z'_t(d)} f(d, w) \cdot \log \frac{N_t}{df_t(w)} \cdot g\left(\log \frac{N_{e,w}}{ef(w)}\right) \tag{17}$$

where $Z'_t(d)$ is a normalization constant, $N_{e,w}$ is the number of events in ROI r that maximized equation (16), and g is a scaling function. We currently use a linear scaling

$$g(x) = (x - A)\frac{D - C}{B - A} + C \tag{18}$$

with $A = \min_w \log(N_{e,w}/ef(w))$, $B = \max_w \log(N_{e,w}/ef(w))$. We experimented with different values for C and D. Currently, $C = 0.8$ and $D = 1.0$ yield the best results, i.e., event-unspecific terms are at most downweighted by a factor of 0.8, while event-specific terms receive their original TF-IDF weight.

4.5 Matching Parts of Documents

Two documents may only partially overlap, even though they are on the same event. This may be due, for example, to additional events being described in one of the documents, or to new information on the same event that is contained in one of the documents but not in the other.

Ideally, we would like to perform topic-based text segmentation on the documents [7, 3] and then compare topically coherent segments of the current document to those of other documents.

In order to create a system in time for the NIST Topic Detection and Tracking evaluation, we used and evaluated a simpler algorithm. Each document is divided into overlapping segments by a sliding window of fixed length l (in words), that advance with a step size s. The sliding window approach is shown in figure 2. If a document is shorter than or contains exactly l words, it consists of exactly one segment, otherwise it consists of two or more segments.

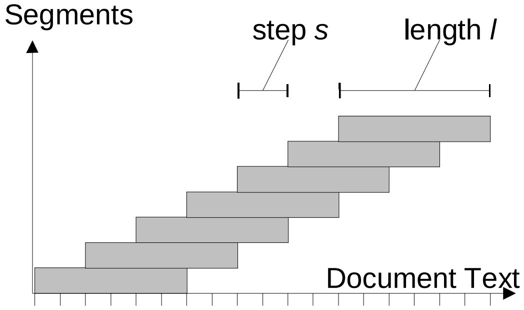

Figure 2: Division of a document into fixed-sized segments of length *l* words and with step size *s* words.

When comparing two documents q and d, we calculate the similarity score of each segment in one document to each segment in the other document. The maximum is used as the overall score:

$$sim_{seg}(q, d) = \max_{s_1 \in q, s_2 \in d} sim''(s_1, s_2) \tag{19}$$

where s_1 and s_2 are the segments in q and d, and sim'' is the similarity score as introduced in equation (15).

As we will see in the experiments section, this simple “tiling” approach improves results. We expect additional benefits from a true topic-based segmentation.

5. EXPERIMENTS AND RESULTS

We used the New Event Detection system described in this paper to participate in the NIST Topic Detection and Tracking evaluation. The system was developed using the TDT2 and TDT3 corpora, where TDT2 served as the training set and TDT3 as a development test set. It was finally tested on the new TDT4 corpus by NIST. In the following, we shortly describe the data sets and then present development and evaluation results.

5.1 Data Sets

The TDT2 corpus consists of 6 months of news stories from the period January to June 1998. It contains approx. 60,000 stories from the text sources Associated Press and New York Times, video broadcasts sources CNN, and ABC, and radio broadcast sources Voice of America, and Public Radio International. We used the automatically speech-recognized and transcribed versions of the TV and radio broadcasts. The total size of the corpus is approx. 35 million words.

The TDT3 corpus consists of 3 months of news stories from the period October to December 1998, containing approx. 37,000 English stories. In addtion to the sources used for TDT2, it contains stories from NBC and MSNBC TV broadcasts. Mandarin stories also contained in the corpus as well as stories in the “TDT3 supplement” were not used for this study.

The TDT4 corpus consists of 4 months of news stories from the period October 2000 to January 2001. It contains approx. 28,000 English stories that were used for the evaluation. The Mandarin and Arabic stories were not used. The English sources are the same as for TDT3.

TDT2 and TDT3 are labeled with 120 topics. Approx. 15,000 English stories belong to one of these topics, the other approx. 45,000 English stories are unlabeled. Topics occurring in TDT2 were used for training, those occuring in TDT3 were used for development.

TDT4 is currently labeled with 40 topics; 20 additional topics are in preparation. These labels were not accesible to us during development.

5.2 Evaluation Metric

TDT uses a cost function C_{Det} that combines the probability of missing a new story p_{miss}, the probability of seeing a new story in the data p_{target}, the cost of missing a new story C_{miss}, the probability of a false alarm p_{FA}, the probability of seeing an old story $p_{nontarget}$, and the cost of a false alarm C_{FA} in the following way:

$$C_{Det} = C_{Miss} \cdot P_{Miss} \cdot P_{target} + C_{FA} \cdot P_{FA} \cdot P_{nontarget} \quad (20)$$

This cost is normalized such that a perfect system scores 0 and a trivial system, the better of always emitting yes or always emitting no, scores 1:

$$(C_{Det})_{Norm} = \frac{C_{Det}}{\min(C_{Miss} \cdot P_{target}, C_{FA} \cdot P_{nontarget})} \quad (21)$$

The system output consists of two parts. The first part is a yes/no decision for each test document, indicating whether the document describes a new event (yes) or a previously seen event (no). This yes/no output is used to determine the normalized cost of the system and also used to compare different systems.

The second part is a confidence score, varying between some minimum and maximum values, were a higher value indicates higher confidence that the document describes a new event. This score is used to plot DET curves for the systems, i.e., curves that plot false alarm vs. miss probabilities. It is also used to determine the minimum normalized cost of the system, i.e., the cost that one would achieve if the optimal threshold on the score were chosen.

A more detailed description of the task and evaluation metric can be found in the TDT-2002 evaluation plan [9].

5.3 Results

We ran our system on the TDT3 and TDT4 data sets. When testing on TDT3, we used TDT2 for training, when testing on TDT4 (the official TDT-2002 evaluation data), we used TDT2 and TDT3 for training.

Table 1 shows topic-weighted and story weighted minimum normalized costs for our systems on the TDT3 dataset. No heldout data set for fine-tuning the threshold θ_s was available for these experiments. We therefore only report minimum costs for our system. The bottom line (line 9) shows results for the base system using the cosine metric. It includes the steps that are described in section 3 (i.e., pre-processing, incremental tf-idf, and normalized term weights), but does not exclude short documents. Line 8 shows results when using Hellinger distance instead of cosine distance. Hellinger performs much better, improving the results by 0.0686. A large number of current IR systems use the cosine distance[4]. This result shows that it is worth exploring other metrics. One hypothesis is that Hellinger distance decreases the effect of outliers in the term counts and weights, but the exact reason for the effect is currently unknown.

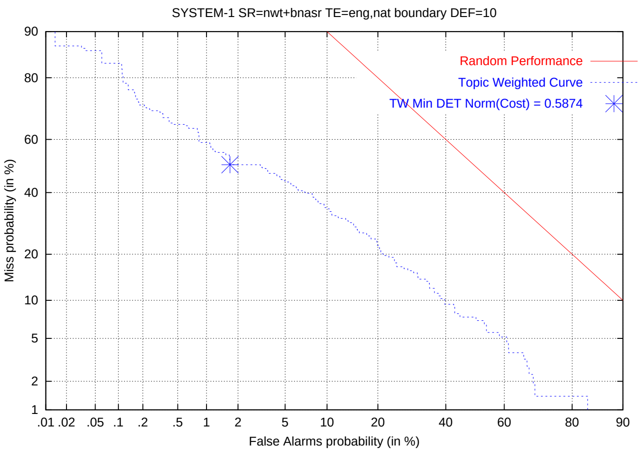

Figure 3: DET curve for SYSTEM-1 on TDT3 data. Minimum cost (cost at the optimal threshold) is 0.5874.

Line 7 additionally excludes short documents from comparison (a document is considered short if it contains less than 50 tokens), yielding a slight improvement. Lines 6 to 1 incrementally add the techniques described in section 4. Each addition only yields a small improvement, but overall this adds up. The best system (line 1) has a topic-weighted minimum normalized cost of 0.5783, which is better than any other result previously reported for that data set. The second-biggest improvement (after exchanging Hellinger for cosine) is the addition of source-pair normalization (line 4; cf. section 4.3), improving results by 0.0333. Figure 3 shows the DET curve for our best system SYSTEM-1, indicating a minimum normalized cost of 0.5834 at a false alarm rate of 0.0170 and a miss rate of 0.5000. The optimal threshold is $\theta_s = 0.8418$. The thresholds at the minimum costs found in this set of experiments were used in the following set of experiments.

Systems labeled SYSTEM-1 to SYSTEM-4 in table 1 (but without having the feature "noshort" turned on) were submitted to TDT-2002. The systems use the following features:

SYSTEM-1 All features used in SYSTEM-2, SYSTEM-3, and SYSTEM-4.

SYSTEM-2 Same as SYSTEM-4; additionally using inverse event frequencies (cf. section 4.4).

SYSTEM-3 Same as SYSTEM-4; additionally matching parts of documents (cf. section 4.5).

[4] Most of these systems do not use a linear term-frequency function as we do. The logarithm is widely used instead. However, in initial experiments we found slightly worse results when using $\log(tf)$ instead of tf for both Cosine and Hellinger distance.

Table 1: New Event Detection Normalized Costs and Minimum Normalized Costs on the TDT3 data set (TDT-2001 evaluation)

		topic-weighted	story-weighted
	System	Min Norm Cost	Min Norm Cost
1.	(SYSTEM-1) base hel noshort source docnorm pairnorm tile eventfreq	**0.5783**	**0.6019**
2.	(SYSTEM-2) base hel noshort source docnorm pairnorm eventfreq	0.5802	0.6127
3.	(SYSTEM-3) base hel noshort source docnorm pairnorm tile	0.5828	0.6091
4.	(SYSTEM-4) base hel noshort source docnorm pairnorm	0.5846	0.6163
5.	base hel noshort source docnorm	0.6179	0.6390
6.	base hel noshort source	0.6264	0.6349
7.	base hel noshort	0.6346	0.6454
8.	base hel	0.6389	0.6483
9.	base cos	0.7075	0.7227

Table 2: New Event Detection Normalized Costs and Minimum Normalized Costs on the TDT4 data set (TDT-2002 evaluation).

	topic-weighted		story-weighted	
System	Norm Cost	Min Norm Cost	Norm Cost	Min Norm Cost
SYSTEM-1	**0.5691**	0.5303	0.5483	0.5133
SYSTEM-2	0.5728	0.5461	0.5444	0.5328
SYSTEM-3	0.5728	0.5287	0.5600	0.5100
SYSTEM-4	0.5957	0.5657	0.5761	0.5411

SYSTEM-4 The basic model (cf. section 3), a source-specific tf-idf model (cf. section 4.1), the average similarity of the current document (cf. section 4.2), and the source-pair-specific average of same-event story similarities (cf. section 4.3).

The official evaluation results are shown in table 2. All systems were run on the TDT4 data without having prior access to the true labels. Normalized costs and minimum normalized costs for our four systems are shown. In the official evaluation metric, topic-weighted normalized cost, our best system (SYSTEM-1) scored second of four submissions[5] with a normalized cost of 0.5691. When evaluating on the story-weighted normalized cost, SYSTEM-2 achieved a normalized cost of 0.5444, which is slightly better than SYSTEM-1.

When testing the topic-weighted minimum-normalized detection cost, SYSTEM-3 (using tiling but not event frequencies) was best with a score of 0.5287. This means, we had a better estimation of the threshold for SYSTEM-1 (tiling plus event frequencies) and SYSTEM-2 (event frequencies), but overall tiling seems to be more valuable than the ad-hoc use of event frequencies.

Our hypothesis is that results are slightly improved when excluding short documents ("noshort"). This feature was not implemented at the time of the TDT submission, and we cannot test this hypothesis since NIST has not published the labeled evaluation data yet. However, running SYSTEM-1 to SYSTEM-4 without the "noshort" option on the TDT-3 data set increases the topic-weighted minimum detection costs to 0.5874, 0.5873, 0.5914, and 0.5932, respectively (compare to lines 1 – 4 in table 1), thus enabling "noshort" yields a minimum cost improvement between 0.007 and 0.010.

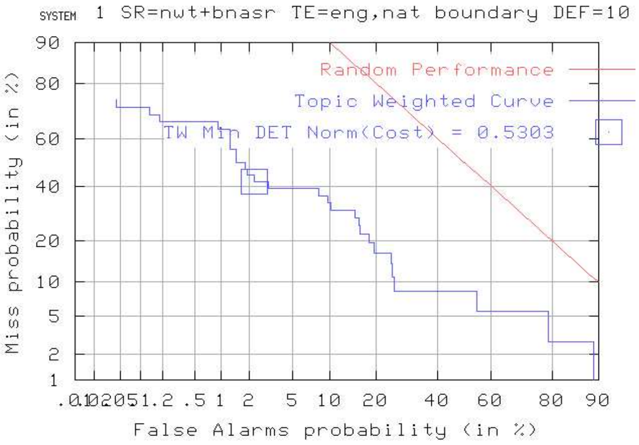

Figure 4: Topic-weighted DET curve for SYSTEM-1 on TDT4 data in the TDT-2002 evaluation. Minimum cost (cost at the optimal threshold θ_s) is 0.5303.

Figure 4 shows the DET curve for SYSTEM-1. The topic-weighted minimum normalized cost is 0.5303 at a false alarm rate of 2.26% and a miss rate of 41.67%. Table 3 shows topic-weighted miss and false alarm rates for the four systems. The false alarm and miss rates indicate that further improvement of the methods are required in order to arrive at a useful stand-alone new event detection system.

[5] TDT participation rules do not allow the publication of results for systems submitted by other groups.

Table 3: Topic-weighted false alarm and miss rates on the TDT4 data set.

System	Norm Cost	False Alarm	Miss
SYSTEM-1	0.5691	3.58%	38.89%
SYSTEM-2	0.5728	3.65%	38.89%
SYSTEM-3	0.5728	3.65%	38.89%
SYSTEM-4	0.5957	3.55%	41.67%

Table 4: Effect of deferral period on minimum normalized detection costs for NED on the 2002 dry run data.

Deferral	min Cost	Δ	% Change
1 file	0.5873	base	–
10 files	0.5883	-0.0010	-0.2%
100 files	0.5930	-0.0057	-1.0%

6. THINGS THAT DID NOT HELP

There were some techniques that we had expected to improve the performance of our systems, but for our implemntation the performance did not improve. These include the use of stories in a "lookahead" when updating the document counts, and the use of the difference in time between two documents in computing a decision score.

6.1 Look-Ahead

The look-ahead that a TDT system is allowed to see (also called the deferral period) can be 1, 10, or 100 files[6]. The incremental TF-IDF model as described in section 3.2 is built from previous stories plus all stories in the deferral period. Table 4 shows the effect on NED cost when changing the deferral period and keeping the rest of the system the same.

Surprisingly, results are best for a deferral period of one file, and worse for longer deferral periods, although the differences in cost are small. The exact reason for this effect is not known yet. However, we hypothesize that it stems from the different weights that are assigned to new terms. With a look-ahead of one file, terms that are new in the current file have a very low document frequency, thus a very high IDF weight. With a longer deferral period, there is a higher chance of seeing that term again in the deferral period, yielding a higher document frequency, thus a lower IDF weight. The lower weight of new terms hurts performance since new words are usually a good indicator of new events.

While there are other ways of using of the deferral period that improve results [4], this experiment indicates that it might be better not to use terms in the deferral period when comparing the current document to documents in the history.

6.2 Using Time Information

We tested two different time decay models. The first one is the IBM exponential time decay model [6]: the similarity of two stories $sim(d,q)$ is adjusted based the difference in time between these two stories (the *age* of the earlier story):

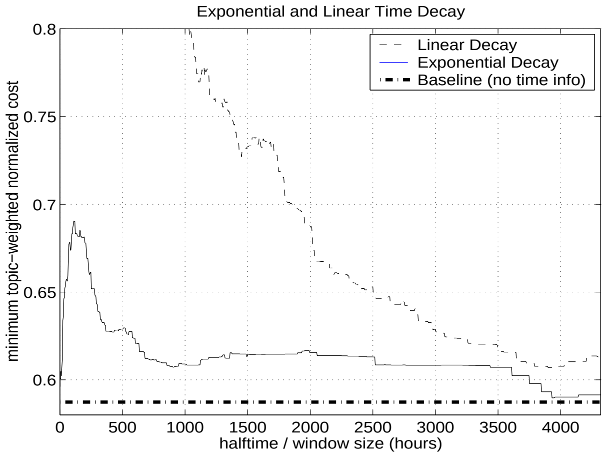

Figure 5: Halftime for exponential time decay and window size for linear time decay vs. topic weighted minimum normalized cost for the NED task.

$$sim^{exp}(d,q) = sim(d,q) \cdot (1{+}2^{-age/halftime}) \qquad (22)$$

Best results on the TDT-2001 evaluation are reported with $halftime$ being around 2 days.

We also tested a linear time model [10]. This model uses a window on the history. The similarity to documents inside this window is

$$sim^{lin}(d,q) = (1 - \frac{age}{m}) \cdot sim(d,q) \qquad (23)$$

where m is the size of the window. The similarity to documents outside this window (i.e., $age > m$) is 0.

Figure 5 shows the baseline result without using time information (horizontal line), and results for halftimes (exponential decay) and window sizes (linear decay) ranging from one hour to 4320 hours (180 days) when training on TDT-2 data and testing on TDT-2002 dry run data. For the exponential model, minimum topic-weighted detection first increases and then approaches the baseline; for the linear model, detection cost starts very high and decreases with larger window sizes and also approaches the baseline. All time-based results are worse than the baseline not using time information.

Our negative findings on the utility of time may be partly explained by the way that we computed the time difference was different from the way IBM computed the time difference [6]. In the IBM model, the time difference is measured as the difference in time between the test story and the first story about an event. The seed story was identified as the first story in an automatically determined cluster about an event. Since our model follows the finding by UMass and CMU [10] that comparing pairs of stories performs better, we used time between the pairs of stories being compared[7].

Another explanation is that as the system performance improves, time becomes less useful. In our earlier systems,

[6] These counts include the current file; one file contains in average 26 stories.

[7] Another difference between our work and the experiments on linear decay decribed in [10] is that we use the actual time stamps of the stories while they used the sequential story id. However, we found results similar to those in figure 5 when replacing the time stamps with story id's.

we noted that time improved performance, but as we improved our systems, using better preprocessing and source-pair specific normalization, time did not help. For our final systems that we submitted to TDT2002, we did not use time.

7. CONCLUSIONS

We presented details of our systems submitted to the TDT-2002 New Event Detection evaluation. Starting with a base system that employs techniques found in most current state-of-the art systems, we replaced the commonly used cosine distance by Hellinger distance, added a source-specific tf-idf model, document similarity normalization, source-pair specific on-topic similarity normalization, the use of inverse event frequencies, and the matching of parts of documents ("tiling"). Each additional element only yields a slight improvement, but taken together we found an improvement from 0.7075 to 0.5783 in topic-weighted minimum normalized detection cost (18% improvement). In the official evaluation, our best system was ranked second of four participants.

We also presented two techniques that we expected to improve results but which did not. These are the use of vocabulary in the look-ahead data for the tf-idf model, and the use of time information. The latter is in contrast to earlier publications.

Two interesting future research directions are the incorporation of event frequencies in a more principled way, and the investigation of other time models that better exploit the time information.

Acknowledgements

Thanks to Kris Popat, Prateek Sarkar, and Ioannis Tsochantaridis for discussions on New Event Detection, to Lauri Karttunen for his spelled-out numbers conversion tool, and to three anonymous reviewers for their comments.

8. REFERENCES

[1] J. Allan, V. Lavrenko, D. Malin, and R. Swan. Detections, bounds, and timelines: Umass and tdt-3. In *Proceedings of Topic Detection and Tracking Workshop (TDT-3)*, Vienna, VA, 2000.

[2] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of SIGIR-98*, pages 37–45, Melbourne, Australia, 1998.

[3] T. Brants, F. Chen, and I. Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *International Conference on Information and Knowledge Management (CIKM)*, McLean, VA, 2002.

[4] J. Carbonell, Y. Yang, R. Brown, C. Jin, and J. Zhang. Cmu tdt report. Slides at the TDT-2002 meeting, CMU, 2002.

[5] W. B. Croft, S. Cronen-Townsend, and V. Larvrenko. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

[6] M. Franz, A. Ittycheriah, J. S. McCarley, and T. Ward. First story detection: Combining similarity and novelty-based approaches. Slides at the TDT-2001 meeting, IBM, 2001.

[7] M. A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[8] V. Lavrenko, J. Allan, E. DeGuzman, D. LaFlamme, V. Pollard, and S. Thomas. Relevance models for topic detection and tracking. In *Proceedings of HLT-2002*, San Diego, CA, 2002.

[9] NIST. The 2002 topic detection and tracking (TDT-2002) task definition and evaluation plan. Technical Report Version 1.1, National Institute of Standards and Technology, 2002.

[10] Y. Yang, T. Pierce, and J. Carbonell. A study on retrospective and on-line event detection. In *Proceedings of SIGIR-98*, Melbourne, Australia, 1998.

[11] Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, 2002.

[12] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of SIGIR-02*, pages 81–88, Tampere, Finland, 2002.