# Classification Models for New Event Detection

Giridhar Kumaran, James Allan and Andrew McCallum
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01003, USA

{giridhar,allan,mccallum}@cs.umass.edu

## ABSTRACT

New event detection (NED) involves monitoring news streams to detect the stories that report on new events. In this paper we explore the application of machine learning classification techniques for this task. We introduce the concept of *triangulation* with illustrative examples. We develop new features that build on this concept, and the named entities present in a document. The classifiers we developed showed significant and consistent improvement over the baseline vector space model system, on all the collections we tested on. Analysis of the performance of our classifiers suggests the utility of named entities, and the applicability of machine learning techniques to the NED task.

## 1. INTRODUCTION

The Topic Detection and Tracking (TDT) program, a DARPA funded initiative, seeks to develop technologies that search, organize and structure multilingual news-oriented textual materials from a variety of broadcast news media. One of the tasks under this program, New Event Detection (NED), requires the constant monitoring of streams of news stories and identifying the first story on *topics* of interest. A *topic* is defined as "a seminal event or activity, along with directly related events and activities"[1]. An earthquake at a particular place could be an example of a topic. The first story on this topic is the story that first carries the report on the earthquakes' occurrence. The other stories that make up the topic are those discussing the death toll, the rescue efforts, the reactions from different parts of the world, scientific discussions, the commercial impact and so on. A good NED system would be one that correctly identifies the article that reports the earthquakes' occurrence as the first story.

NED systems are very useful in situations where novel information needs to be ferreted out from a mass of rapidly growing data. Examples of real-life scenarios are financial markets, news analyses, intelligence gathering etc.

Stories reporting new events are detected by comparing them with all the stories that have arrived in the news stream in the past. Metrics such as cosine similarity, Hellinger similarity[5], KL divergence etc. are used for determining how closely related two stories are. The most effective contemporary systems are those that build on the vector space model and cosine similarity metric. Previous attempts at using language modeling techniques for NED haven't been able to better the results obtained by vector space models. Similarly, attempts to use machine learning techniques haven't paid off as well. These attempts at different models and alternative approaches are to a large extent due to the fact that performance improvements the can be realized from the vector space model systems have plateaued.

In this paper, we view NED as a binary classification problem - i.e., each story has to be classified into one of two categories - *old* or *new*. While there are are a number of classification models available, we experimented with support vector machines (SVMs). Our attempts to apply these machine learning classification techniques to the NED problem achieved considerable success. We attribute this more to the innovative features we incorporated than the actual classification models themselves.

This paper begins by summarizing the previous work in NED in Section 2. We then briefly describe the evaluation methodology for NED in Section 3. Section 4 discusses the importance of feature selection, and provides an overview of the two classfication models we experimented with. In Section 5 we detail our feature selection process and introduce our new concept of triangulation with illustrative examples. While Section 6 describes the experimental setup and the pre-processing we did to the data, Section 6 provides information on our baseline NED system. We then describe the model creation process in Section 7 and provide the results of applying these models in Section 8. The results are analyzed in Section 8.1. We finally wrap up with conclusions and future work in Section 9.

## 2. PREVIOUS RESEARCH

On-line NED was the focus of a paper by Papka et al[12]. When a new document was encountered, it was processed immediately to extract features and build up a query representation of the document's content. The document's initial threshold was determined by evaluating it with the query. If the document did not trigger any previous query by exceeding this particular threshold, it was marked as a new event. The threshold model developed for the task incorpo-

*CIKM'04,* Submission. Do not cite or distribute.

rated time information, the intuition being that documents that are widely spaced apart in time are more likely to deal with new (different) events. Performance-wise, it was found that increasing the number of features used to build the queries results in improved performance, with an unacceptable increase in running time of the system. At low feature dimensionality, misses were attributed to the inability of the feature extraction process to weight event-level features more heavily than more general topic-level features. Even at higher feature dimensionalities misses occurred, which were finally ascribed to the poor weight assignment strategy for query features.

A paper by Stokes et al. [14] presented an approach to NED that utilized a combination of evidence derived from two distinct representations of a document's content. While one of the representations was the usual free text vector, the other made use of lexical chains (created using WordNet) to obtain the most prevalent topics discussed in the document - again as a vector of terms. This method automatically disambiguated terms. The two vectors were combined in a linear fashion, and the usual cluster-document similarity-threshold approach was followed. It was concluded that a marginal increase in effectiveness could be achieved when lexical chain representations are used in conjunction with the free text representation, i.e. the data fusion model was marginally better.

Allan et al.[3] argued that NED approaches that relied on exploiting existing news tracking technology would invariably exhibit poor performance. Systems that used tracking technology for NED followed the mantra - every time a new topic was found and tracked by a topic tracking system, it was equivalent to finding a new event. Thus, the NED system was only as good as the tracking system it was built on. Given tracking error rates, the lower and upper bounds on NED error rates were derived mathematically. These values were found to be good approximations of the true NED system error rates. Since tracking and filtering using full-text similarity comparison approaches were not likely to make the sort of improvements that are necessary for high-quality NED results, the paper concluded that an alternate approach to NED was required.

A summer workshop[2] on topic-based novelty detection held at Johns Hopkins University extensively studied the NED problem. Similarity metrics, effect of named entities, pre-precessing of data, and language and hidden markov models were explored. Combinations of NED systems were also discussed.

In the topic-conditioned novelty detection[16] approach, documents were classified into broad topics and NED was performed within these categories. Additionally, named entities were re-weighted relative to the normal words for each topic, and a stop list was created for each topic. However the experiments were done on a corpus different from the TDT corpus.

Brants et al. [5] extended a basic incremental TF-IDF model to include source-specific models, similarity score normalization techniques, and segmentation of documents. Good improvements on TDT benchmarks were shown.

The most recent paper on NED by Kumaran et al[11]. introduces better document models and similarity metrics by leveraging the utility of named entities. Stories were classified into different categories and category-specific stop words were removed. Once this was done, three different similarity metrics were calculated between each pair of stories. These corresponded to the overlap of all the terms in the documents, only the named entities in the documents, and finally only the non-named entities in the document. The first of the scores mentioned was modified based on the category the story belonged to as well as the other two scores, and reported as the "newness" confidence score for that story. In this paper we utilize the new similarity metrics that were developed and use them as features to build formal models for NED.

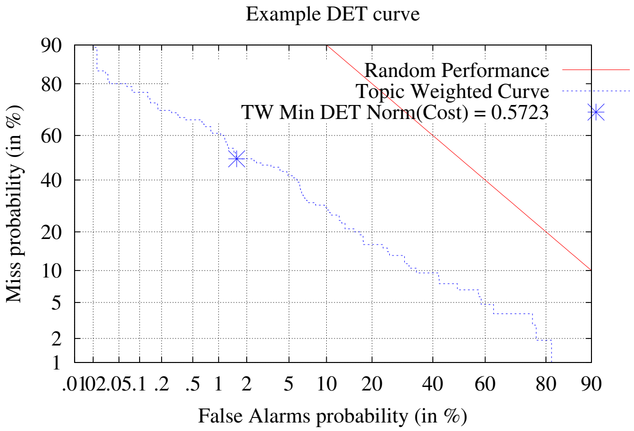

**Figure 1: An example DET curve. Each point on the curve is the due to the misses and false alarms at a particular threshold. A sweep across all the possible thresholds from 0 to 1 generates the points in the DET curve.**

## 3. NED EVALUATION

The official TDT evaluation requires a NED system to assign a confidence score between 0 and 1 to every story upon its arrival. This assignment of scores is done either immediately upon arrival or after a fixed look-ahead window of stories. A (cosine similarity) score of 0 translates to complete confidence that the story is new, and a score of 1 implies the greatest confidence that the story is old. To evaluate performance, the stories are sorted according to their scores, and a threshold sweep is performed. All stories with scores above the threshold are declared old, while those below it are considered new. At each threshold value, the misses and false alarms are identified, and a cost is calculated as a linear function of their number. The threshold that results in the least cost is selected as the optimum one. Different NED systems are compared based on their minimum cost. The Detection Error Tradeoff (DET) curve is a convenient way to represent the miss and false alarm values at each threshold, and to compare the performance of NED algorithms at different regions of the graph, i.e. at different thresholds. Figure 1 is an example DET curve.

## 4. FEATURE EXTRACTION AND CLASSIFICATION MODELS

The goal of feature extraction is to obtain a representation of an object so that objects in the same category have similar representative values while objects in every other different category have very different representative values.

Classification models use the representations provided by the feature extractor to classify an object into one of many categories. Usually, it is impossible to classify an object strictly into a single category. Hence, most classification models provide a membership probability for the object for each category. A simple example is the Bayesian classifier that provides a posterior probability as a discriminant function to map an object to its class.

## 4.1 Support Vector Machines

Support Vector Machines are margin classifiers, i.e. they attempt to generate a hyper-plane, which is the discriminant function, that separates two classes of training examples with the largest margin [6]. The maximum margin is desirable as it leads to better classifier performance. The hyper-plane is constructed in a higher dimensional space called kernel space, which is mapped to from the feature space.

### *4.1.1 Kernel Functions*

SVMs use kernel functions that allow us to operate in the input feature-space while providing us the ability to compute inner products in the kernel space. The key idea in mapping to a higher space is that, in a sufficiently high dimension, data from two categories can always be separated by a hyper-plane. Examples of kernel functions are linear, polynomial, string, radial basis function, and sigmoid kernels.

# 5. FEATURE SELECTION

The length of a story, its source, the number of named entities, the names of people involved in the story, the time of occurance, the language it is in, etc., are naive features that are simply of no use in a task as complex as NED. This is so because pinning down the character of new stories is a tough process. New events don't follow any periodic cycle, can occur at any instant, can involve only one particular type of named entity or a combination, can be reported in any language, and can be reported as a story of any length by any source[1]. Hence we decided that the best features to use would be those that were not particular to the story in question only, but those that measure differences between the story and all the stories it is compared with.

Kumaran et al. [11] developed category-specific rules that modified the baseline confidence score assigned to each story based on the overlap of named entities and non-named entities with the closest story reported by a baseline system. We decided to use these three scores: namely, the baseline confidence score, named entity overlap, non-named entity overlap along with the category the story was assigned to as the four features to start with. The named entities considered were *Event*, *GPE*, *Language*, *Location*, *Nationality*, *Organization*, *Person*, *Cardinal*, *Ordinal*, *Date*, and *Time*.

## 5.1 Triangulation

Every news story is characterized by a set of named entities and a set of terms that discuss the topic of the story. We refer to the latter as *topic terms*. For an old story, there would be significant overlap of both the named entity terms as well as topic-terms with some story seen in the past (see following example). However this will not be true for a new story. Intuitively, a new story can atmost share one of either the named entity terms or the topic terms (if it shared both the named entities as well as the topic terms with a single story, then it has to be old) with a single story. This also impies that the story that shares the same named entities with a new story must be different from the story that shares the same topic terms. Further, these two stories must themselves be on different topics, i.e. they should have very low similarity. We call this concept *triangulation*.

We now illustrate the triangulation concept for old and new stories with examples.

### *5.1.1 An old story*

**Story 1 : Old Story**

*While in **Croatia** today,**Pope John Paul II** called on the **international community** to **help** end the fighting in the Yugoslavia's **Kosovo** province.*

**Story 2 : Closest Match**

***Pope John Paul II** is urging the **international community** to quickly **help** the ethnic Albanians in **Kosovo**. He spoke in the coastal city of Split, where he ended a three-day visit to **Croatia**.*

**Story 1** is an *old* story about Pope John Paul II's visit to Yugoslavia. **Story 2** was the first story on the topic and it shares both named entities likes **Pope John Paul II** and **Croatia** and also topic-terms like **international community** and **help**. Thus we see that for *old* stories both the named entities as well as topic-terms overlap (measured by means of cosine similarity) with either the same story or very similar stories.

### *5.1.2 A new story*

**Story 3 : New Story**

***Turkey** has sent 10,000 troops to its southern border with Syria amid growing tensions between the two neighbors, newspapers reported Thursday. Defense Minister **Ismet Sezgin** denied any troop movement along the border, but said **Turkey's** patience was running out. **Turkey** accuses Syria of harboring Turkish Kurdish rebels fighting for autonomy in **Turkey's** southeast; it says rebel leader Abdullah Ocalan lives in Damascus.*

**Story 4 : Closest Story due to Named Entities**

*A senior **Turkish** government official called Monday for closer military cooperation with neighboring Bulgaria. After talks with President Petar Stoyanov at the end of his four-day visit, Turkish Deputy Premier and National Defense Minister **Ismet Sezgin** expressed satisfaction with the progress of bilateral relations and the hope that Bulgarian-**Turkish** military cooperation will be promoted.*

**Story 3** is a *new* story about the rising tensions between Turkey and Syria. The closest story due to named entity overlap as reported by our baseline cosine similarity system is **Story 4**, a story about Turkish-Bulgarian relations.

[1] It could be argued that articles from a source, say*NYTimes*, are much longer than news stories from *CNN*, and hence the length of stories is a good candidate for use as a feature. However, when there is no pattern that indicates that either of the two sources reports new stories preferentially, the use of length as a feature is moot.

The named entities **Turkey** and **Ismet Sezgin** caused this match.

**Story 3 : New Story**
*Turkey has sent 10000* ***troops*** *to its* ***southern*** *border with Syria amid growing tensions between the two neighbors, newspapers* ***reported*** *Thursday. Defense* ***Minister*** *Ismet Sezgin* ***denied*** *any* ***troop*** *movement along the border, but said Turkey's patience was running out. Turkey accuses Syria of harboring Turkish Kurdish* ***rebels fighting*** *for autonomy in Turkey's southeast; it says* ***rebel*** *leader Abdullah Ocalan lives in Damascus.*

**Story 5 : Closest Story due to Topic Terms**
*Sudanese* ***troops*** *are chasing Ugandan and Eritrean forces out of Sudan after crushing them in battles in the* ***south*** *of the country, the interior* ***minister*** *said. Sudan has claimed that heavy clashes have been taking place in* ***southern*** *Sudan since Sept. 14, when it said Ugandan and Eritrean* ***troops*** *entered Sudan to support local* ***rebel*** *groups. Uganda and Eritrea have* ***denied*** *the reports. Christian and Animist rebels have been* ***fighting*** *a military campaign against the Khartoum government since 1983 for more autonomy for the southern districts.*

The closest story due to topic term overlap as reported by our baseline cosine similarity system is **Story 5**, a story on a completely different topic. Terms like **minister**, **fighting**, **rebel**, etc., caused this match.

Thus we see that for new stories, the named entities and topic terms match with different stories, and these stories themselves have a low similarity between themselves. We now have three more features: namely, cosine similarity with the story 'A' whose named entities match most closely, cosine similarity with the story 'B' whose topic terms match most closely, and cosine similarity between the stories 'A' and 'B' themselves. It is obvious from our discussion on old stories that 'A' and 'B' can be the same story for such stories [2].

We summarize the features in Table 1 and present a pictorial representation of the feature selection process in Figure 2.

## 6. EXPERIMENTAL SETUP

We used the TDT2, TDT3 and TDT4 data sets for our experiments. The TDT2 corpus covers the period from January 4 to June 30, 1998 and has around 70,000 stories from New York Times, Associated Press Worldstream, CNN, ABC, PRI and VOA. TDT3 contains news stories from October to December 1998. It contains around 35,000 stories from sources like CNN, New York Times, ABC, Voice of America etc. TDT4 consists of approximately 28,500 stories from the period October 2000 to January 2001, and from the same sources. Only the English stories in the collection were considered. TDT2 contains 100 topics (and hence hundred new events), TDT3 115 topics , while TDT4 contains 70 topics. We used TDT3 stories to train our classification models to be used on TDT2, and stories from both TDT2 and TDT3 for our tests on TDT4.

To develop SVM models we used $SVM^{Light}$[8], which is an implementation of SVMs in C. $SVM^{Light}$ is an implementation of Vapnik's Support Vector Machine [15] for the problems of pattern recognition, regression, and learning a ranking function. The optimization algorithms used in $SVM^{Light}$ are described in [9] and [8].

We used version 1.9 of the open source Lemur system[3] to tokenize the data, remove stop words, stem and create document vectors. We used the 418 stopwords included in the stop list used by InQuery [7], and the K-stem stemming algorithm [10] implementation provided as part of Lemur.

Incremental TF-IDF weighting[5] was used, and document similarity normalization [5] was performed before a final score was assigned to a story.

We used the cosine similarity (Equation 1) metric to judge the similarity of a story with those seen in the past. The documents are represented as term vectors with TF-IDF weighting. The maximum similarity of the story with stories seen in the past is taken as the confidence score that the story is old.

$$Sim(d,d') = \frac{\sum_w weight(w,d) * weight(w,d')}{\sqrt{\sum_w weight(w,d)^2}\sqrt{\sum_w weight(w,d')^2}} \tag{1}$$

where

$$weight(w,d) = tf * idf$$

$$tf = \log(termfrequency + 1.0)$$

$$idf = \log((docCount + 1)/(documentfreq + 0.5))$$

This constituted our baseline system.

## 7. BUILDING CLASSIFICATION MODELS

We used the features mentioned in Section 5 to build SVM models. As a baseline case, we built SVM models using MS (the original cosine similarity) as the only feature. The SVM model, upon testing, exactly matched the performace of our baseline system. The category feature ($c$) was dropped as it inhibited performance. We believe that its utility was confined to the manually constructed decision rules like those in Kumaran et al.[11]. Apparently, SVMs were unable able to learn such intricate decision rules. We provided combinations of the remaining features as input and built models using linear, polynomial, and RBF kernels.

## 8. RESULTS AND ANALYSIS

We used the models trained on different collection to test the appropriate collection as indicated in Section 6. The official conditions for running our systems as well as the nature of the data to work on laid down by the TDT program were followed. We found that using certain kernels and certain combinations of features improved performance over the baseline system significantly. We found that results for all three corpura, TDT2, TDT3, and TDT4, were

[2] Triangulation breaks down when the old story is the second story in a topic and 'A' and 'B' are different stories. In that case 'A' and 'B could have low similarity and we would get a spurious feature value.

[3] http://www.cs.cmu.edu/~lemur

**Table 1: A summary of the features extracted for the NED task for a story$S$.**

| Symbol | Feature |
|---|---|
| $c$ | Category of the story [11] determined using BoosTexter[13] |
| MS | Original baseline-system cosine similarity with a story, say $X$ |
| MS-ne | Cosine similarity between only the named entities in $S$ and $X$ |
| MS-top | Cosine similarity between only the topic terms in $S$ and $X$ |
| NE-Sim | Cosine similarity with story $Y$ whose named entities match most closely with those in $S$ |
| Top-Sim | Cosine similarity with story $Z$ whose topic terms match most closely with those in $S$ |
| Triang-Sim | Cosine similarity between $Y$ and $Z$ |

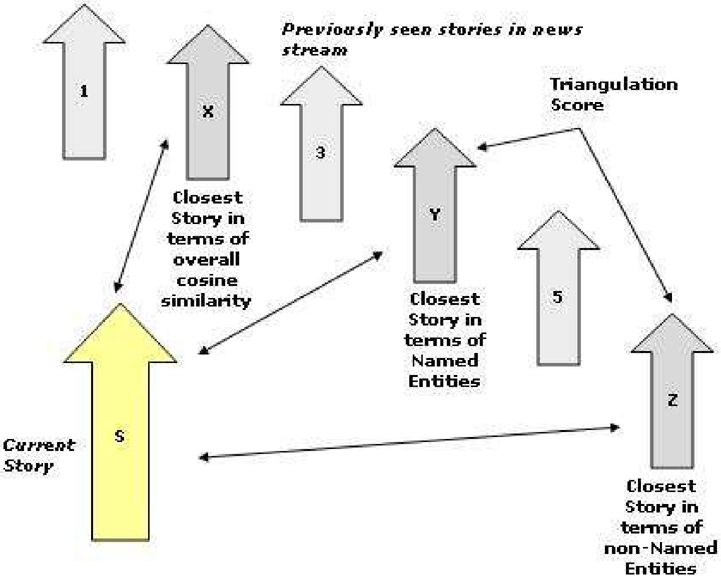
Previously seen stories in news stream
1
X
3
Y
5
Z
S
Triangulation Score
Closest Story in terms of overall cosine similarity
Closest Story in terms of Named Entities
Closest Story in terms of non-Named Entities
Current Story

**Figure 2: Story 7 is the new story in this example. Story 2 is used to calculate MS,MS-ne, and MS-top feature values. Story 4 and Story 6 are used to obtain NE-Sim and Top-Sim feature values respectively. The simlarity between Story 4 and Story 6 is the value of the feature Triang-Sim.**

**Table 2: Summary of the results of using SVM classification models for NED on the TDT2, TDT3, and TDT4 collections. The results for the TDT3 collections were obtained by training on TDT3 itself, and hence are at best suggestive.**

| Features | Kernel Type | TDT2 | TDT3 | TDT4 |
|---|---|---|---|---|
| **Baseline System** | | **0.5885** | **0.5744** | **0.6673** |
| MS,MS-ne | Radial Basis Function | 0.5599 | 0.5377 | 0.6110 |
| MS,MS-ne,MS-top | Radial Basis Function | 0.5076 | 0.5687 | 0.6522 |
| MS,MS-ne,MS-top,NE-Sim | Radial Basis Function | 0.5735 | 0.515 | 0.6573 |
| MS,MS-ne,MS-top,NE-Sim,Top-Sim | Radial Basis Function | 0.527 | 0.5442 | 6858 |
| MS,MS-ne,MS-top | Polynomial of degree 2 | 0.5258 | 0.5757 | 0.6536 |
| MS,MS-ne,MS-top | Polynomial of degree 3 | 0.5317 | 0.5681 | 0.6496 |
| MS,MS-ne,MS-top,NE-Sim,Top-Sim,Triang-Sim | Polynomial of degree 3 | 0.5697 | 0.5157 | 0.6614 |

**Table 3: Summary of the results of using SVM classification models for NED on the TDT4 ASR and close-captioned (CC) versions of the collections.**

| Features | Kernel Type | TDT4 ASR | TDT4 CC |
|---|---|---|---|
| **Baseline System** | | **0.5144** | **0.6673** |
| MS,MS-ne | Radial Basis Function | 0.6795 | 0.6110 |
| MS,MS-ne,MS-top | Radial Basis Function | 0.7829 | 0.6522 |
| MS,MS-ne,MS-top,NE-Sim | Radial Basis Function | 0.7828 | 0.6573 |
| MS,MS-ne,MS-top,NE-Sim,Top-Sim | Radial Basis Function | 0.8076 | 0.6858 |
| MS,MS-ne,MS-top | Polynomial of degree 2 | 0.6892 | 0.6536 |
| MS,MS-ne,MS-top | Polynomial of degree 3 | 0.6968 | 0.6496 |
| MS,MS-ne,MS-top,NE-Sim,Top-Sim,Triang-Sim | Polynomial of degree 3 | 0.8334 | 0.6614 |

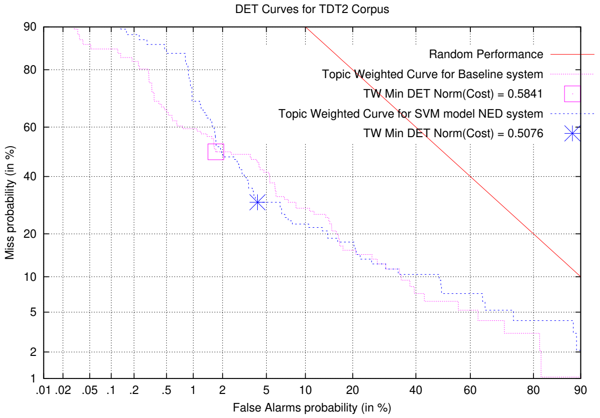

**Figure 3: DET curves for the TDT2 collection.**

consistently and significantly improved by using the classification models. Table 2 summarizes the results we obtained. The baseline system we used was the state-of-the-art system available. The numbers presented in the table are the minimum cost values (Section 3).

We used the close-captioned verions of the three corpora for our experiments. In order to try our systems on a different type of corpus, we ran the system on the automatic speech recognition (ASR) version of TDT4. The results of doing so are provided in Table 3. We noticed that all our SVM model-based NED systems performed worse than the baseline system.

## 8.1 Analysis

The main goal of our effort was to come up with some way to correctly identify new stories based on some features we thought characterized new stories. To understand what we had actually achieved by using these models we studied the distribution of the scores of new stories and old stories for the baseline and SVM model-based NED systems.

The distributions of scores for new and old stories for the baseline system and a SVM model-based NED systems on TDT2 are presented as Figure 4 and Figure 5 respectively. The corresponding DET curves are presented in Figure 3

We observe that the scores for a small fraction of new

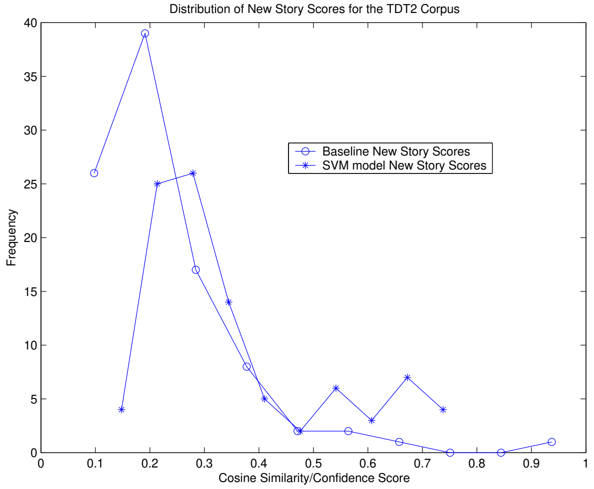

**Figure 4: Distribution of new story scores for the baseline and SVM model systems.**

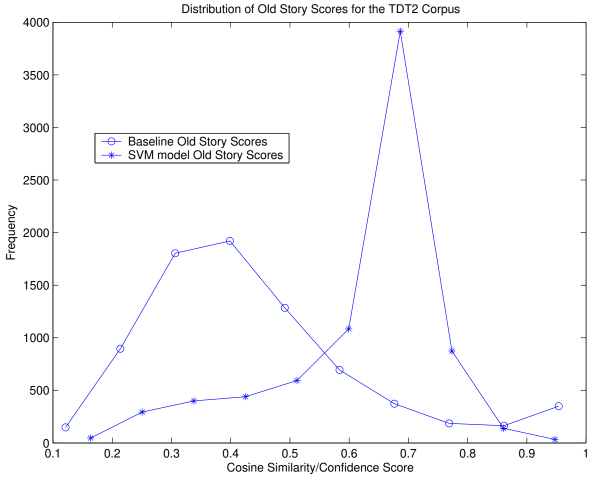

**Figure 5: Distribution of old story scores for the baseline and SVM model systems.**

stories are reduced by the model-based NED system while a larger fraction is increased by a small amount. However, the major impact of using SVM model-based NED systems appears to be in detecting old stories. We observe that the scores of a significant number of old stories (compared to new stories) have been increased to be closer to one. This had the effect of increasing the score difference between old and new stories, and hence improved classification accuracy as measured by the minimum cost.

While there was no single SVM model that always outclassed other models in performance, we observed that RBF models with various combinations of features on an average performed better that linear and polynomial kernels. The results we obtained were also better than the best performance reported in the most recent published work on NED by Kumaran et al[11].

## 8.2 ASR corpus

While the performance on the close-captioned corpora was more than satisfactory, the performance on the ASR version was not good. However, analysis of the TDT4 ASR results revealed an interesting aspect of the problem. The success of our classification models depended not only on the selection of good features, but also on how well the values of these features were estimated. A majority of the features (and their values) depended on the accurate identification of the named entities in the corpus. We used the publicly available BBN Identifinder [4] to do so. For TDT4, the TDT program stipulated that a part of the corpus include stories that had been converted from speech to text by automatic speech recognition(ASR) systems. The text of such stories does not include punctuation and is completely in upper case - clues used by BBN Identifinder to identify named entities. Hence BBN Identifinder failed to identify named entities correctly, and all the estimates of our feature values were wrong. This was the reason why our classification models performed poorly on TDT4.

# 9. CONCLUSIONS AND FUTURE WORK

We have shown the applicability of machine learning classification techniques to solve the NED problem. Significant improvements were made over the baseline systems on all the corpora tested on except the ASR version. The features we engineered made extensive use of named entities, and reinforced the importance and need to effectively harness their utility to solve problems in TDT. The problems encountered with ASR documents are further testimony to the importance of named entities. From the study of the distributions of scores assigned to stories by the baseline and SVM model systems, we believe that attacking the problem as "*old story detection*" might be a better and more fruitful approach.

For future work, engineering of better features is a definite priority. Since NED systems are expected to work with input in any form, the problem associated with ASR documents needs to be addressed. One option is to re-train BBN Identifinder [4] on ASR documents.

## Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

# 10. REFERENCES

[1] J. Allan. *Topic Detection and Tracking: Event-Based Information Organization*. Kluwer Academic Publishers, 2002.

[2] J. Allan, H. Jin, M. Rajman, C. Wayne, G. D., L. V., R. Hoberman, and D. Caputo. Summer workshop final report. In *Center for Language and Speech Processing*, 1999.

[3] J. Allan, V. Lavrenko, and H. Jin. First story detection in tdt is hard. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pages 374–381, 2000.

[4] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.

[5] T. Brants, F. Chen, and A. Farahat. A system for new event detection. In *Proceedings of ACM SIGIR 2003*, pages 330–337, 2003.

[6] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[7] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY retrieval system. In *Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications*, pages 78–83, 1992.

[8] T. Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 169–185. MIT Press, 1998.

[9] T. Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, 2002.

[10] R. Krovetz. Viewing morphology as an inference process. In *Proceedings of ACM SIGIR93*, pages 61–81, 1998.

[11] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *Proceedings of ACM SIGIR2004*, 2004.

[12] R. Papka and J. Allan. On-line new event detection using single pass clustering TITLE2:. Technical Report UM-CS-1998-021, , 1998.

[13] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. In *Machine Learning 39(2/3):1*, pages 35–168. Kluwer Academic Publishers, 2000.

[14] N. Stokes and J. Carthy. First story detection using a composite document representation. In *Proceedings of Human Language Technology Conference*, 2001.

[15] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[16] Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In *Proceedings of ACM SIGKDD03*.