Topic-conditioned Novelty Detection

Yiming Yang, Jian Zhang, Jaime Carbonell, Chun Jin
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-8213, USA
{yiming,jianzhan,jgc,cjin}@cs.cmu.edu

ABSTRACT

Automated detection of the first document reporting each new event in temporally-sequenced streams of documents is an open challenge. In this paper we propose a new approach which addresses this problem in two stages: 1) using a supervised learning algorithm to classify the on-line document stream into pre-defined broad topic categories, and 2) performing topic-conditioned novelty detection for documents in each topic. We also focus on exploiting named-entities for event-level novelty detection and using feature-based heuristics derived from the topic histories. Evaluating these methods using a set of broadcast news stories, our results show substantial performance gains over the traditional one-level approach to the novelty detection problem.

Categories and Subject Descriptors

I.5.2 [**Design Methodology**]: Classifier design and evaluation; Feature evaluation and selection; Pattern analysis;; H.3.3 [**Information Search and Retrieval**]: Information filtering

General Terms

Design, Experimentation, Algorithm

Keywords

Novelty Detection, Named Entity, Feature Selection, Text Classification

1. INTRODUCTION

Automated detection of new events from chronologically ordered documents (e.g., newswire stories or TV broadcasts) is an open challenge in text mining. A specific form of novelty detection, namely First Story Detection (FSD), is the task of online identification of the earliest report for each event as soon as that report arrives in the sequence of documents. FSD has been recognized as the most difficult task in the research area of Topic Detection and Tracking (TDT), compared to the other tasks like *known event tracking* and *retrospective event detection*. Current FSD systems are mostly based on comparing a new document to all the documents in the past, and thresholding on the similarity scores – if all the similarity scores are below a threshold, the new document is predicted as the first story of a novel event. Such a simple-minded approach yielded limited performance in TDT benchmark evaluations. A performance upper-bound analysis by Allan et al.[2] provided a probabilistic justification for the observed performance degradation in FSD compared to event tracking, and suggested that new approaches must be explored in order to significantly enhance the current performance level achieved in FSD.

In this paper, we focus on how to use training data of old events to learn useful statistics for the prediction of new events. More specifically, we investigate a new approach consisting of the following components:

1. classifying documents into broad topics each of which consists of multiple events;
2. identifying Named Entities (NE's), optimizing their weight relative to normal words for each topic, and computing a stopword list per topic;
3. measuring the novelty of a new document conditioned on the system-predicted topic for that document.

The rationale behind our approach is that events belonging to the same topic often share a set of keywords. For example, documents talking about different events in *airplane accidents* (as a topic) tend to share the words like "airplane", "crash" and "accidents". Those keywords are informative for discriminating on-topic and off-topic documents, but, they also make the first story of a new airplane crash look just like the documents for another airplane crash which has already occurred, and cause FSD systems to miss such reports unless the system has the ability to distinguish the *features* (words or phrases) discriminative for *topic classification* and those discriminative for *event distinction*. To obtain such functionality, we propose the approach of topic-conditioned FSD in which we use supervised learning algorithm to classify documents by topic, and topic-conditioned feature weights to measure the novelty of documents at event level.

Section 2 describes our method; Section 3 introduces the data and performance measures for evaluation, and the similarity matrices illustrating the confusibility among events within each topic; Section 4 reports experimental results;

SIGKDD 02 Edmonton, Alberta, Canada

and Section 5 summarizes the research findings and discusses issues for further research.

2. A NEW SCHEME FOR FSD

To avoid ambiguities in terminology, let us make a distinction between the words *topic* and *event*. By event we mean that some action happened during certain time period and at certain location, e.g., the *TWA-800 crash*; by topic we mean a recurring and broader class of events, e.g., *airplane accidents*. An event includes a set of documents, and a topic consists of its children events [1].

2.1 The Two-level Scheme

Figure 1 shows the two-level scheme for FSD conditioned on topics. The first level is for the classification of documents by topic, using a supervised learning algorithm[10]. The second level consists of several identical copies of an FSD algorithm: each is in charge of FSD for a particular topic.

The FSD algorithm is extremely simple: when a new document arrives, it is compared to all the documents in its past ("the history"). If its nearest neighbor in the past has a cosine similarity score (or any reasonable choice of similarity measure) below a threshold, then the new document is labeled as "NEW", meaning it is the first story of a novel event; otherwise, it is labeled as "OLD". After this, the document is added to the history. The threshold is empirically set based on cross validation. This approach has been typical among current FSD systems in the TDT benchmark evaluations, and has been reported in detail in previous papers[9, 13]. When using the FSD algorithm in our two-level FSD scheme, the only modifications we made are that we use one FSD for each topic, and that each FSD keeps its *local history* (and the derived statistics) instead of the *global history* in a traditional FSD system. That is, documents are automatically routed to the corresponding topic by the classifier at the first level before they are sent to the second level for novelty detection.

The two-level scheme allows us to treat documents and words in different ways at each level. For topic-level classification, we would like to give more weight to the topical discriminative words, like "airplane", "accident", "tornado", "hijacking", etc., while at the novelty detection level,

we can also generate a stopword list for each topic based on how common a word is used in that topic, and give more weight to the event-level discriminative terms, like "TWA-800" or "September 11th", for example.

2.2 Topic-specific Stopword Removal

Since topical common words cause events in the same topic to be mutually confusing, and are a potential cause for a FSD system to miss the first story of a new event, a natural choice for us is to remove those words. We obtained a stopword list for each topic by thresholding on the training-set document frequency of a word:

$$\frac{n(t, T_i)}{n(T_i)} > \beta \tag{1}$$

where $n(T_i)$ is the number of documents on topic T_i; $n(t, T_i)$ is the number of documents containing word t and on topic T_i; and parameter β is empirically chosen by running the two-level scheme FSD system on a validation corpus.

[1]Events, not topics, have been the central foci in TDT; however, for some historical reason, topic and event have been used interchangeably in the TDT literature.

2.3 Weighted Use of Named Entities

In this study we used BBN's Hidden Markov Model software [4] which extracts seven types of general-purpose Named Entities (NEs), including *Person*, *Organization*, *Location*, *Date*, *Time*, *Money* and *Percent*. Intuitively, NEs would be informative for differentiating events, although their effects in automated FSD have not been examined. Our questions here are: which types of NEs are particularly useful for topic-conditioned FSD, and how can we effectively use them?

We apply NE identification to every document when it is routed to the second level of the FSD system for a specific topic, and used these extracted NEs to extend the vocabulary of that topic [2]. The vector representation of a document for that topic is defined to be:

$$\vec{d} = (w_{1,d}, w_{2,d}, \ldots, w_{|V|,d}, u_{1,d}, u_{2,d}, \ldots, u_{|S|,d}) \tag{2}$$

$$w_{i,d} = (\log(1 + n(t_i, d))) \cdot \log(\frac{N}{n(t_i, D)}) \tag{3}$$

$$u_{j,d} = \alpha_k \cdot (\log(1 + n(s_j, d))) \cdot \log(\frac{N}{n(s_i, D)}) \tag{4}$$

where

- N is the total number of documents in the *local history*
- $w_{i,d}$ is the within-document weight for word t_i in the current vocabulary V (which is adaptive over time); $u_{j,d}$ is the within-document weight for Named Entity $s_i \in S$, and S is the set of NEs extracted by far;
- $n(t_i, d)$ is the within-document term frequency (TF) of word $t_i \in V$; $n(s_j, d)$ is the within-document NE frequency of NE $s_j \in S$;
- $n(t_i, D)$ is the within-collection document frequency of term t_i and D is the collection of documents processed by far; $n(s_j, D)$ is the the within-collection document frequency of NE s_j;
- α_k is the weight of the k^{th} type of NEs, empirically tuned through validation.

2.4 Topic-sensitive Feature Weighting

Here *feature* means either a term or a named entity. In the following experiments, terms are stemmed, and stopwords are removed. We aim to use topic-sensitive weights of features to improve both the topic classification part of our approach and the topic-conditioned novelty detection part. For the first part, we want to choose topic-level discriminative features, and for the second part, we want a heuristic indicator for the usefulness of certain types of NEs.

We applied the χ^2 average criterion [12] to feature weights conditioned on topics, which we have found consistently outperforms other feature selection criteria such as the *information gain* and *document frequency* on several benchmark collections in text classification evaluations. The χ^2 statistics for terms are computed using a training set of documents with topic labels from which a contingency table is derived:

[2]By "extend" we mean that those terms contained in the NEs are still kept in the document.

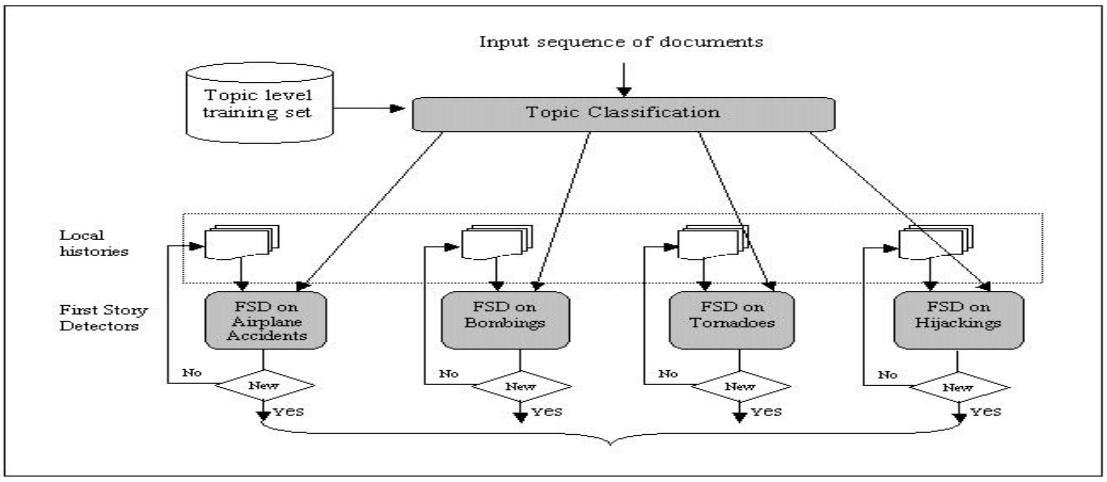

Figure 1: Topic-conditioned First Story Detection (FSD)

Table 1: A two-by-two contingency table

	# of doc's on topic T	# of doc's off topic T
# of doc's containing f	A	B
# of doc's not containing f	C	D

The χ^2 statistic for a specific feature f with respect to topic T is defined to be:

$$\chi^2(f,T) = \frac{(A+B+C+D)\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \quad (5)$$

The average weight for feature f over all the topics is computed using

$$\chi^2_{avg}(f) = \sum_{i=1}^{m} P_r(T_i)\chi^2(f,T_i) \quad (6)$$

Ranking features by their cross-topic averaged scores and thresholding on the ranks yields a selected subset of features for the topic-level classification of documents (the first level in Figure 1).

To estimate the *effectiveness* of each type of Named Entity, we compute the χ^2_{avg} value for each NE and average those values over all the NEs in the same type:

$$E_k = \frac{1}{|S_k|}\sum_{f\in S_k}\chi^2_{avg}(f) \quad (7)$$

where E_k is the effectiveness for the k^{th} type of NEs, and S_k is the set of NEs in that type. In the previous section, we mentioned empirically tuning the weight for each type of NE .The E_k value enables us to estimate the potential effects of different types of NEs without actually running exhaustive validation experiments, and thus applying fine tuning only to the promising types of NEs. As is shown in Section 4, this measure is correlated to the empirical results, and thus is a useful indicator for comparing different types of NEs.

3. DATA AND METRICS

3.1 Data

For empirical examination, ideally, we would like to have a document collection with a large number of manually labeled topics and events, and with a reasonable number of documents for each event. In reality, such data sets are difficult to find. The TDT benchmark collections, for example, have over 300 manually defined events but broad topic labels are not available, unfortunately. We also found that the TDT events are often sparsely labeled for a given topic, e.g., only three bombing events were labeled among many bombing events actually reported in the TDT corpora. Those sparsely labeled events per topic do not allow us to thoroughly examine the power of our method in differentiating mutually confusible events. We therefore created our own data collection for the evaluation.

The source data, named Broadcast News and published by Primary Source Media, consists of 261,209 transcripts for news articles from ABC, CNN, NPR, and MSNBC in the period from 1992 to 1998. Each transcript comes along with a record that is composed of several fields, such as title, date, source, keywords, abstract and body. We only extracted the body field of each record into a "bag of words", and called it a document. Human-assigned keywords are informative for grouping documents that share a certain topic. For the experiments in this paper, we defined four broad topics – "Airplane Accidents", "Bombings", "Hijackings" and "Tornadoes" – and collected the on-topic documents using the corresponding keywords of each topic.

For each topic, we identified a set of events by randomly sampling some documents within that topic and manually defining the events described in those documents. For each event, we created a brief description indicating what it is about, where and when it happened, who was involved, etc. The event definitions were used by humans to assign event labels to documents, but not used by the system. Further, we used the sampled documents for each event as the queries to retrieve similar documents from the pool of the documents in the same topic. The pooled documents were manually labeled with respect to the defined events.

Using this procedure, we intended to define 10 different events for each topic, but, for topic *Hijackings*, we only found 6 events in the data collection. Finally, by taking

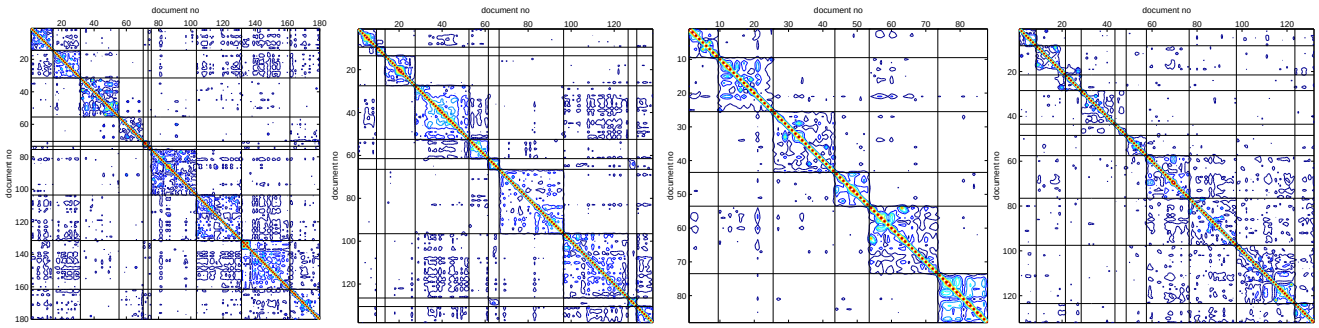

Figure 2: Similarity Matrices: Airplane Accidents, Bombings, Hijackings and Tornadoes

the union of the documents with event labels (defined), we get 538 documents, each exclusively belonging to one of 36 events under 4 topics.

The documents were streamed back together in the original temporal order. The events happened earlier are used as the training set, and the events happened later are used as the test set. The training set was used to train the topic-level classifier, and to tune parameters (the weights for NEs relative to original words) and generate topic-specific stop-word lists. The test set was used for evaluating our system which was tuned in the training phase. The output of our system is a binary decision on each test document in the stream about whether or not it is the first story of an unseen event thus far.

3.2 Similarity Matrices

Before running experiments on this data set, we would like to analyze how difficult the corpus is. That is to say, to analyze how confusible the events in this corpus are for a system making distinctions. We define a *similarity matrix* for the documents in each topic as below:

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1N} \\ S_{21} & S_{22} & \cdots & S_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ S_{N1} & S_{N2} & \cdots & S_{NN} \end{bmatrix} = \begin{bmatrix} S^{11} & S^{12} & \cdots & S^{1M} \\ S^{21} & S^{22} & \cdots & S^{2M} \\ \cdots & \cdots & \cdots & \cdots \\ S^{M1} & S^{M2} & \cdots & S^{MM} \end{bmatrix} \quad (8)$$

where

- N is the number of documents in a particular topic; M is the number of events in the topic;
- documents $\{\vec{d_1}, \vec{d_2}, \cdots, \vec{d_N}\}$ are sorted *by event* as the first key and *by the time of arrival* as the second key;
- S_{ij} is the cosine similarity between document vectors $\vec{d_i}$ and $\vec{d_j}$; S^{kl} is the sub-matrix consisting of the inter-event similarity scores between events k and l when $k \neq l$, or the intra-event similarity scores otherwise.

Clearly, S is a symmetric matrix and $S_{ii} = 1$. Such a matrix enables us to visualize how difficult it is for a similarity-based system to make distinctions among events in the same topic. We used Matlab to draw contours on the similarity scores, making it clear where the dense parts are in a similarity matrix. Figure 2 demonstrates the *similarity matrices* of four topics respectively. From these graphs we can see that although the diagonal sub-matrices are more dense than off-diagonal sub-matrices, the values in those off-diagonal sub-matrices are not negligible, indicating the difficulty in separating inter-event stories.

3.3 Evaluation Measures

To evaluate the performance of our system, we choose the conventional measures for FSD used in the TDT benchmark evaluations [1]. Those measures are defined to be:

$$C_{fsd} = C_{miss} \cdot P_{miss} \cdot P_{target} + C_{fa} \cdot P_{fa} \cdot P_{non-target} \quad (9)$$

$$(C_{fsd})_{norm} = \frac{C_{fsd}}{min(C_{miss} \cdot P_{target}, C_{fa} \cdot P_{non-target})} \quad (10)$$

where

- C_{miss} and C_{fa} are the costs of a *miss* and a *false alarm*, we use $C_{miss} = 1.0$ and $C_{fa} = 0.1$, respectively,
- P_{miss} and P_{fa} are the conditional probabilities of a *miss* and a *false alarm*, respectively,
- P_{target} and $P_{non-target}$ are the a prior target probabilities ($P_{non-target} = 1 - P_{target}$). In TDT benchmark evaluations, P_{target} was set to 0.02 for all events; here we follow the same convention.

If the system classifies a true first story of some event as "No", it commits a miss error. If the system classifies a non-first story as "Yes", it commits a false alarm error. P_{miss} is the ratio of the number of miss errors to the number of the first stories (= number of events) in the stream. P_{fa} is the ratio of the number of false alarm errors to the total number of non-first stories. Miss errors are considered as a more severe problem than false alarm, as people may miss earliest reports on important events. This justifies the higher value of the cost C_{miss}.

The normalized cost $(C_{fsd})_{norm}$ computes the relative cost of the system with respect to the minimum of two trivial systems (one simply makes "Yes" decisions and another simply makes "No" decisions without examining the stories).

Finally, the normalized cost can be either computed for the system's decisions over the cross-product of all stories and all events (called "story-weighted" or "micro-averaged"), or computed for each event separately, then averaged over events (which is called "topic-weighted" or "macro-averaged").

4. EXPERIMENTS

4.1 Topic-level Classification

For upper-level part of our system, we chose a Rocchio-style classifier which is one of the state-of-the-art systems

in the TDT benchmark evaluations for event tracking[5, 11] and the TREC evaluations for adaptive filtering[3]. The Rocchio method was originally developed for query expansion using relevance feedback in text retrieval[6, 7]. Applied to text classification, it computes a *prototype* vector for each class as a weighted average of positive and negative training examples. The prototype for topic T_j is defined to be:

$$\vec{p}_j(\gamma, n) = \frac{1}{|\mathcal{D}(T_j)|} \sum_{\vec{d}_i \in \mathcal{D}(T_j)} \vec{d}_i - \gamma \frac{1}{|\mathcal{D}_n(\bar{T}_j)|} \sum_{\vec{d}_j \in \mathcal{D}_n(\bar{T}_j)} \vec{d}_j \quad (11)$$

where $\vec{d}$ is a training document; $\mathcal{D}(T_j)$ is the set of positive training examples of topic T_j; $\mathcal{D}_n(\bar{T}_j)$ is the "query zone"[8], consisting of the n top-ranking documents retrieved from the negative training examples when using the positive centroid (the first term in the formula) as the query; and γ is the weight of the negative centroid. The use of a query zone is an important departure from the original Rocchio algorithm and is intended to deal with the larger sets of negative examples available in text classification. Detailed description about this classifier can be found in [11].

Applying Rocchio to our experiments, we set the parameters $\gamma = 1.0$, and $n = 400$. We also applied feature selection before running Rocchio using the χ^2_{avg} criterion, resulting in 400 selected features. The topic classificaiton performance is .85 micro-averaged F1 (the harmonic average of recall and precision)[10] and .84 macro-averaged F1.

4.2 Main Results

We conducted four experiments under different conditions, and summarize their condidions and results in Table 2 [3].

Table 2: Summary of Experimental Results

Cases	Micro-avg C_{fsd}	Reduced Cost(%)	Macro-avg C_{fsd}	Reduced Cost(%)
baseline	0.5498	——	0.5353	——
simple case	0.5332	-3.02%	0.5245	-2.02%
ideal case	0.3551	-35.41%	0.3786	-29.27%
real case	0.4548	-17.28%	0.4515	-15.65%

For the *Baseline*, we ran our one-level FSD system on the training set to tune the "novelty threshold", and applied the tuned FSD method to the test set.

Simple Case is the same as *Baseline* except that it is the two-level system with perfect topic labels. In other words, we used human-assigned topic labels to route each document to the corresponding FSD system at the second level, and then ran the FSD system conditioned on that topic. From the results we can see that there is no big difference between results of *baseline* and *simple case*, meaning that the topic label alone is of little additional value for this corpus.

For the *Ideal Case*, we again used perfectly-assigned topic labels, but also applied both the NE weighting and removed topic-specific stopwords. Assuming perfect topic classification, of course, only gives us the performance upper-bound of the two-level approach; nevertheless, it provides a chance to examine the net effect of NEs and feature heuristics in our new two-level FSD approach.

[3]The C_{fsd} we use here and later is the normalized cost introduced in section 3.

For the *Real Case*, we used the Rocchio-style classifier (micro-averaged F1 performance = .85) instead of perfectly-assigned topic labels, but the condition is otherwise identical to the ideal one. We see that we still gain a major performance improvement over the baseline, though not as much as the ideal case. Classifier errors account for the difference in performance.

4.3 NE Weighting

Based on the application of NE discussed above, we first calculated the *effectiveness* (defined in equation 7) of seven types of NEs for each topic, and the results are showed in Table 3.

Table 3: Effectiveness of different types of NEs

NE Type	Effectiveness of seven types of NEs			
	AIR	**BOMB**	**HIJ**	**TORN**
Location	**2.61**	**3.02**	**3.77**	**2.15**
Person	1.56	1.78	1.46	1.34
Organization	2.02	1.79	2.09	1.36
Time	1.24	1.03	1.13	1.07
Date	2.31	2.27	1.37	2.18
Money	2.43	2.10	0.92	1.11
Percent	1.08	0.73	1.01	0.96
mean of all features	2.29	2.12	1.82	1.40

Using the *effectiveness* as a indicator, we can safely reduce the parameters' search space. As shown in Table 3, "Location" is the most informative type of NEs, we then decided to treat that type of NEs as one group, and the remaining six types of NEs as another group. After we tuned the weights (α in formula 4) for these two groups using the training set, we got a weight of 4.0 for the "Location" type of NEs and 1.0 for the other six types of NEs.

To be more clear regarding how much confusibility has been reduced by our approach, we show two similarity matrices for the topic *Airplane Accidents* here (each one contains the five events in the test set), one for *baseline* and one for *ideal case* (figure 3) [4].

From the color graphs we can see that in the ideal case matrix, not only do off-diagonal sub-matrices become more sparse, but the diagonal sub-matrices also become more dense.

5. CONCLUSIONS AND FUTURE WORK

By applying a new approach to FSD, we effectively reduced the degree of confusibility between events within each topic and gained a substantial performance improvement in novelty detection. Our study shows that the topic-conditioned novelty detection approach allows better exploitation of named entities and feature-based heuristics in representing topic histories, indicating clear promise of the approach and inviting further research.

[4]The difference between the baseline matrix here and the one we showed in section 3 is: we used "perfect" *idf* (computed retrospectively from the whole corpus) in section 3; while here we didn't take training set as part of the *adaptive idf*, which is computed as each test document comes in. As a result, the matrix in section 3 looks more dense, i.e. confusible, than the baseline matrix showed here.

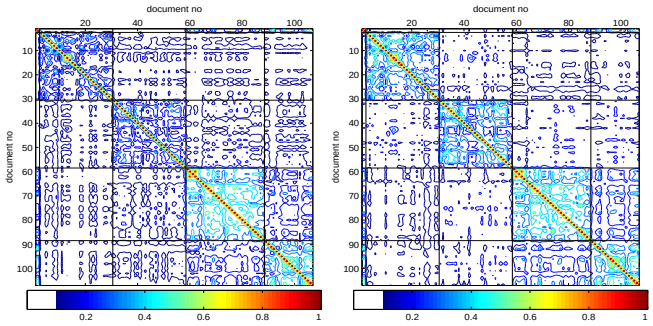

Figure 3: Similarity Matrices of Airplane Accidents: baseline and ideal case

Several important areas need to be further studied in the future, including:

1. Situated role extraction for NEs: While we have shown the usefulness of NEs in topic-conditioned FSD, we believe situated NEs would far more powerful. By "situated" NEs, we mean NEs plus their roles under context, e.g., "TWA 800" as the flight number in an airplane crash. Automated induction of Finite State Transducers for the extraction of "situated" NEs is a challenging research topic.

2. Automated hierarchical clustering: Instead of using human defined topics at the upper-level in our scheme, using system-generated clusters is another challenging problem.

3. Adaptive learning at the topic classification level: Substantial developments have been made recently in the adaptive filtering area of information retrieval[3], and investigating those new techniques for novelty detection has not been done.

6. ACKNOWLEDGMENTS

We thank Charles Wayne from DoD for his guidance in the TDT task definition and evaluation. We also thank Fan Li who helped to create some runs of the topic-level classification. This research is sponsored in part by National Science Foundation (NSF) under the grant number KDI-9873009, and in part by NSF under the grant number IIS-9982226. However, any opinions or conclusions in this paper are the authors' and do not necessarily reflect those of the sponsors.

7. REFERENCES

[1] The 2001 topic detection and tracking (tdt2001) task definition and evaluation plan. In *http://www.nist.gov/speech/tests/tdt/tdt2001/evalplan.htm*, 2001.

[2] J. Allan, V. Lavrenko, and H. Jin. First story detection in tdt is hard. Washiongton DC, 2000. Proceedings of the Ninth International Conference on Informaiton and Knowledge Management (CIKM).

[3] T. Ault and Y. Yang. knn, rocchio and metrics for information filtering at trec-10. In *Proceedings of TREC-10*, 2002 (to appear).

[4] D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning named-finder. In *In Fifth Conference on Applied Natural Language Processing*, 1997.

[5] J. Fiscus, G. Doddington, J. Garofolo, and A. Martin. Nist's 1998 topic detection and tracking evaluation (tdt2). In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 19–26, San Francisco, CA, 1999. Morgan Kaufmann Publishers, Inc.

[6] J. J. Rocchio-Jr. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Inc., Englewood Cliffs, New Jersay, 1971.

[7] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of American Society for Information Sciences*, 41:288–297, 1990.

[8] R. E. Schapire, Y. Singer, and A. Singhal. Boosting and rocchio applied to text filtering. In *Proceedings of the Twenty-first Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 215–223, New York, 1998. The Association for Computing Machinery.

[9] F. Walls, H. Jin, S. Sista, and R. Schwartz. Topic detection in broadcast news. In *Proceedings of the DARPA Broadcast News Workshop*, pages 193–198, San Francisco, CA, 1999. Morgan Kaufmann Publishers, Inc.

[10] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.

[11] Y. Yang, T. Ault, and T. Pierce. Combining multiple learning strategies for effective cross validation. In *The Seventeenth International Conference on Machine Learning (ICML'00)*, pages 1167–1182, 2000.

[12] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In J. D. H. Fisher, editor, *The Fourteenth International Conference on Machine Learning (ICML'97)*, pages 412–420. Morgan Kaufmann, 1997.

[13] Y. Yang, T. Pierce, and J. Carbonell. A study on retrospective and on-line event detection. In *Proceedings of the 21th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 28–36, 1998.