A Probabilistic Model for Retrospective News Event Detection

Zhiwei Li
Microsoft Research Asia
Beijing, China 100080
t-zli@microsoft.com

Bin Wang*
University of Science and Technology of China
Hefei, China, 230027
binwang@ustc.edu

Mingjing Li, Wei-Ying Ma
Microsoft Research Asia
Beijing, China 100080
{mjli,wyma}@microsoft.com

ABSTRACT

Retrospective news event detection (RED) is defined as the discovery of previously unidentified events in historical news corpus. Although both the contents and time information of news articles are helpful to RED, most researches focus on the utilization of the contents of news articles. Few research works have been carried out on finding better usages of time information. In this paper, we do some explorations on both directions based on the following two characteristics of news articles. On the one hand, news articles are always aroused by events; on the other hand, similar articles reporting the same event often redundantly appear on many news sources. The former hints a generative model of news articles, and the latter provides data enriched environments to perform RED. With consideration of these characteristics, we propose a probabilistic model to incorporate both content and time information in a unified framework. This model gives new representations of both news articles and news events. Furthermore, based on this approach, we build an interactive RED system, HISCOVERY, which provides additional functions to present events, *Photo Story* and *Chronicle*.

Categories and Subject Descriptors

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

General Terms

Algorithms, Management

Keywords

Retrospective News Event Detection, Maximum Likelihood, Expectation Maximization, Clustering

*This work was performed when the second author was a visiting student at Microsoft Research Asia.

SIGIR'05, August 15–19, 2005, Salvador, Brazil.

1. INTRODUCTION

A news event is defined as a specific thing happens at a specific time and place [1], which may be consecutively reported by many news articles in a period. Retrospective news Event Detection (RED) is defined as the discovery of previously unidentified events in historical news corpus [12]. RED has lots of applications, such as detecting earthquakes happened in the last ten years from historical news articles.

Although RED has been studied for many years, it is yet an open problem [12]. A news article contains two kinds of information: contents and timestamps. Both of them are very helpful for RED task, but most previous research work focus on finding better utilizations of the contents [12]. The usefulness of time information is often ignored, or at least time information is used in unsatisfied manners. According to these observations, we explore RED from the following two aspects. On the one hand, we consider the better representations of news articles and events, which should effectively model both the contents and time information. On the other hand, we notice that the previous work consider little on modeling events in probabilistic manners. As a result, in this paper we propose a probabilistic model for RED, in which both contents and time information are utilized. Furthermore, based on it, we build a RED system, HISCOVERY (HIStory disCOVERY), which can provide a vivid multimedia representation of the results of event detection. Our main contributions include:

1). Proposing a multi-modal RED algorithm, in which both the contents and time information of news articles are modeled explicitly and effectively.

2). Proposing an approach to determine the approximate number of events from the articles count-time distribution.

The remainder of this paper is organized as follows. We review the previous research work in section 2. In section 3, we study the characteristics of news articles and news events. The proposed approach is presented in section 4. In section 5, we briefly introduce our RED system, HISCOVERY. Then we report on experimental methodologies and results in section 6 and 7. At last, we conclude our paper and discuss the future plans in section 7 and section 8, respectively.

2. RELATED WORK

RED was firstly proposed and defined by Yang et al. [12], and an agglomerative clustering algorithm (augmented *Group Average Clustering*, GAC) was proposed in that paper, but since then there are few right-on-the-target research work reported. But a similar topic, *New Event De-*

tection(NED), has been extensively studied. It is noted that some researchers use very similar algorithms to perform both NED and RED. Thus, we mainly review the previous work on NED in this section.

The most prevailing approach of NED was proposed by Allan et al. [3] and Yang et al. [12], in which documents are processed by an on-line system. In such on-line systems, when receiving a document, the similarities between the incoming document and the known events (sometime represented by a centroid) are computed, and then a threshold is applied to make decision whether the incoming document is the first story of a new event or a story of some known event. Modifications to this approach may be summarized from two aspects: better representation of contents and utilizing of time information.

From the aspect of utilizing the contents, TF-IDF is still the dominant technique for document representation, and cosine similarity is the generally used similarity metric. However, many modifications have been proposed in recent years. Some work focus on finding new distance metrics, such as the Hellinger distance metric [5]. But more works focus on finding better representations of documents, i.e. feature selection. Yang et al. [11] classified documents into different categories, and then removed stop words with respect to the statistics within each category. Significant improvements were reported by them. The usage of named entities have been studied, such as in Allan et al. [2], Yang et al. [11] and Lam et al. [7], but there are yet no generally acknowledged conclusions on whether named entities are useful. Reweighting of terms is another prevailing method, firstly proposed by Allan et al. in [2]. In [11], Yang et al. proposed to re-weight both named entities and non-named terms with respect to statistics within each category. A recent publication of Kumaran et al. [6] summarized the work in this direction and proposed some extensions. They exploited to use both text classification and named entities to improve the performance of NED. In their work, stop words are removed conditioned on categories, similar with the method of Yang et al. [11], but they relaxed the constraint on document comparison: the incoming document were compared with all documents instead of only documents belonging to the same category. Then each document was represented by three vectors: the whole terms, named entities and non-named entity terms. But there are no consistently best representations of documents for all categories.

From the aspect of utilizing time information, generally speaking, there are two kinds of usages. Some approaches, such as the on-line nearest neighbor approach discussed above, only use the chronological order of documents. The other approaches, such as [12] and [5] use decaying functions to modify the similarity metrics of the contents.

A unique thinking of NED is proposed by Zhang et al. [13], in which the authors distinguished the concepts of relevance and redundancy, and argue that relevance and redundancy should be modeled separately.

3. CHARACTERISTICS OF NEWS ARTICLES AND EVENTS

Although the RED task is well defined, with the booming of the World Wide Web, the environments changed. Thus, we should reconsider RED under the new situations. The most important change is that the number of news articles is increasing dramatically, because: i) the Web provides cheaper and instantaneous platforms to publish news articles, and ii) telecoms networks connect the world more tightly than ever before, which enable more and more events happening faraway to be aware and reported. These tremendous news articles bring some difficulties to RED, but they also give us data enriched environments to perform RED. Moreover, this change pushes the emergence or exposure of some characteristics of news articles and events, which are very helpful for RED. Figure 1 illustrates the count of news stories on topic "Halloween" posted by CNN and MSNBC as a function of time. The horizontal axis is time, and the vertical axis is number of stories. It is worth noticing that, here, "Halloween" is a topic, while it includes lots of events (i.e. each year's Halloween is an event)[1]. This figure indicates the two most important characteristics of news articles and events:

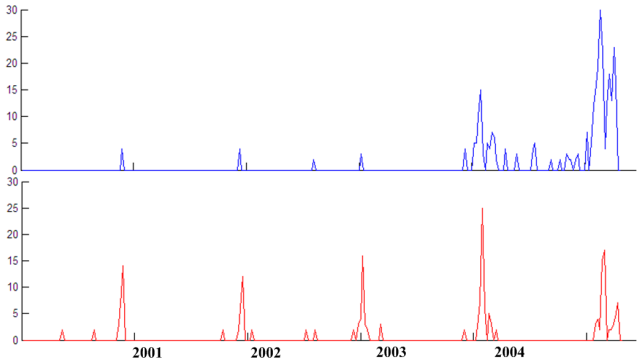

Figure 1: Halloween reported on MSNBC(top) and CNN(bottom). The unit of the horizontal axis is a week. News stories reporting "Halloween" only appear around the Halloween.

1). News reports are always aroused by news events, and the articles counts of an event are changed with time. Mapping to the plot of article count-time distribution, events are peaks(as shown in Figure 1). However, in some situations, several events could be overlapping on time, that is, the observed peaks and events are not exactly corresponding, or peaks could not be observed clearly.

2). Both the contents and time of the articles reporting the same event are similar on different news sites, especially the articles of important events. The start and end time of reports to events on different websites are very similar. Although, the quantities of news articles on different websites are different, their tendencies are very similar. For example, as shown in Figure 1, in each year, both CNN and MSNBC start to report "Halloween" from the beginning of October and stop on the early of December immediately.

These characteristics are very helpful for RED. The first characteristic leads RED problem to be modeled by a latent variable model, where events are latent variables and articles are observations. The second characteristic enables us to gather lots of news stories on the same event by mixing articles coming from different sources together. Because news stories posted on websites are easy to obtain and become more and more important ways to publish news, in this paper, we focus on detecting events from them.

[1]In this paper, we intend to find many events in a set of news stories rather than only one event or topic.

4. MULTI-MODAL RESTOSPECTIVE NEWS EVENT DETECTION METHOD

As mentioned above, both news articles and events could be represented by two kinds of information: contents and timestamps. These two kinds of information have different characteristics, thus, we propose a multi-modal approach to incorporate them in a unified probabilistic framework.

4.1 Representations of News Articles and News Events

According to the knowledge about news, news articles can be further represented by four kinds of information: *who* (persons), *when* (time), *where* (locations) and *what* (keywords). Similarly, a news event also can be represented by persons, time(defined as the period between the first article and the last article), locations and keywords. For news article, the timestamp is a discrete value, while for news event, its time consists of two values. As a result, we define news article and event as:

$$article = \{persons, locations, keywords, time\}$$
$$event = \{persons, locations, keywords, time\}$$

The keywords represent the remainder contents after removing named entities and stop words. The contents of news articles are divided into three kinds of information. In order to simplify our model, we assume the four kinds of information of a news article are independent:

$$p(article) = p(persons)p(locations)p(keywords)p(time)$$

Usually, there are many named entities and keywords in news articles, and we generally term them as entity in this paper. As a result, there are three kinds of entities, and each kind of entity has its own term space.

4.2 The Generative Model of News Articles

According to the first characteristic of news articles and events, the generation processes of news articles can be modeled by a generative model. Since contents and timestamps of news articles are heterogeneous features, we model them with different types of models.

Contents The bag of words model is an effective representation of documents, and the Naïve Bayes(NB) classifier basing on this model works very well on many text classification and clustering tasks [8]. Thus, just like in NB, we use mixture of unigram models to model contents. It is important to note that person and location entities are important information of news articles, but they only take a small part of the contents. If we model the whole contents with one model, this important information may be overwhelmed by keywords. Thus, we model persons, locations and keywords by three models, although as will cause extra computational cost.

Timestamps As mentioned in the previous section, each event corresponds to a peak on articles count-time distribution whether it can be observed or not. In other words, the distribution is a mixture of many distributions of events. A peak is usually modeled by a Gaussian function, where the mean is the position of the peak and the variance is the duration of event. As a result, *Gaussian Mixture Model*(GMM) is chosen to model timestamps.

Consequently, the whole model is the combinations of the four mixture models: three mixture of unigram models and one GMM. Before illustrating the model, let us define the notations used in this paper. The news article, x_i, is represented by three vectors, $persons_i$, $locations_i$ and $keywords_i$, and one timestamp, $time_i$. The vector, $persons_i$, ($locations_i$ and $keywords_i$ are defined similarly) is considered to be a list, $< person_{i1}, \ldots, person_{iN} >$, and each element is the occurrence count of corresponding entity in x_i. The j-th event is represented by e_j. The two-step generating process of a news article is:

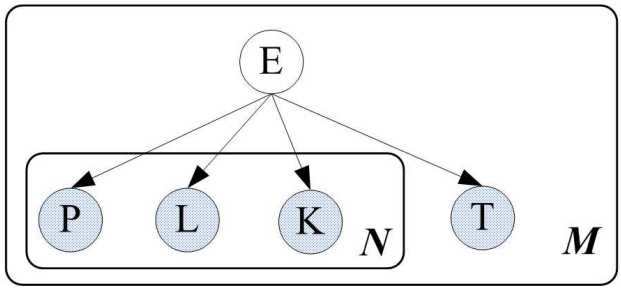

Figure 2: Graphical model representation of the generative model of news articles. E, P, L, K and T represent events, persons, locations, keywords and time respectively. Shadow nodes are observable; otherwise is hidden. N(entities) and M(articles) at the bottom-right corners represent plates.

1. Choose an event $e_j \sim Multinomial(\theta^j)$
2. Generate a news article $x_i \sim p(x_i|e_j)$. For each entity of it, according to the type of current entity:
   a. Choose a person $person_{ip} \sim Multinomial(\theta_p^j)$
   b. Choose a location $location_{ip} \sim Multinomial(\theta_l^j)$
   c. Choose a keyword. $keyword_{ip} \sim Multinomial(\theta_n^j)$

For its timestamp:
   a. Draw a timestamp $time_i \sim N(u^j, \sigma^j)$

where the vector θ^j are mixing proportions, or the priors of events; θ_p^j, θ_l^j, and θ_n^j are parameters of conditional multinomial distributions given event e_j; μ^j and σ^j are parameters of the conditional Gaussian distribution given event e_j. Figure 2 is a graphical representation of this model. In this figure, we use N to represent the term space sizes of the three kinds of entities(N_p, N_l and N_n).

4.3 Learning Model Parameters

The model parameters can be estimated by Maximum Likelihood method. As shown in Figure 2, by introducing latent variable, events, we can write the log-likelihood of the joint distribution as:

$$\begin{aligned} l(X;\theta) &\triangleq \log(p(X|\theta)) = \log(\prod_{i=1}^{M} p(x_i|\theta)) \\ &= \sum_{i=1}^{M} \log(\sum_{j=1}^{k} p(e_j)p(x_i|e_j,\theta)) \end{aligned} \quad (1)$$

where X represents the corpus of news articles; M and k are number of news articles and number of events respectively; is model parameters. Given an event j, the four kinds of information of the i-th article are conditional independent:

$$\begin{aligned} p(x_i|e_j) = p(time_i|e_j)p(persons_i|e_j)p(locations_i|e_j) \\ p(keywords_i|e_j) \end{aligned} \quad (2)$$

Expectation Maximization(EM) algorithm is generally applied to maximize log-likelihood. The parameters could be

estimated by running E-step and M-step alternatively. By using Jensen's inequality and the independent assumptions expressed in (2), in M-step, we can decouple equation (1) into the sum of four items. In each of these four items, there are only parameters of one model. Thus, parameters of the four mixture models can be estimated independently. In E-step, we compute the posteriors, $p(e|x_i)$, by:

$$p(e_j|x_i)^{(t+1)} = \frac{p(e_j)^{(t)}p(x_i|e_j)^{(t)}}{p(x_i)^{(t)}} \propto p(e_j)^{(t)}p(x_i|e_j)^{(t)} \quad (3)$$

where the upper script (t) indicates the t-th iteration. In M-step, we update the parameters of the four model. Since persons, locations and keywords are modeled with independent mixture of unigram models, so their update equations are the same, and we use token w_n to represent the n-th entity. For the three mixture of unigram models, parameters are updated by:

$$p(w_n|e_j)^{(t+1)} = \frac{1 + \sum_{i=1}^{M} p(e_j|x_i)^{(t+1)} * tf(i,n)}{N + \sum_{i=1}^{M}(p(e_j|x_i)^{(t+1)} * \sum_{s=1}^{N} tf(i,s))} \quad (4)$$

where $tf(i,n)$ is the count of entity w_n in x_i and N is the vocabulary size. For each type of entities, N is the size of corresponding term space. Since the co-occurrence matrix is very sparse, we apply Laplace smoothing [8] to prevent zero probabilities for infrequently occurring entities in (4). The parameters of the GMM are updated by:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{M} p(e_j|x_i)^{(t+1)} * time_i}{\sum_{i=1}^{M} p(e_j|x_i)^{(t+1)}} \quad (5)$$

$$\sigma_j^{(t+1)} = \frac{\sum_{i=1}^{M} p(e_j|x_i)^{(t+1)} * (time_i - \mu_j^{(t+1)})^2}{\sum_{i=1}^{M} p(e_j|x_i)^{(t+1)}}$$

It is important to note that because both the means and variances of the Gaussian functions are changed consistently with the whole model, the Gaussian functions work like sliding windows on time line. By this way, we overcome the shortcomings caused by the fixed windows or the parameter-fixed decaying functions used in traditional news event detection algorithms [12][5]. At last, the mixture proportions are updated by:

$$p(e_j)^{(t+1)} = \frac{\sum_{i=1}^{M} p(e_j|x_i)^{(t+1)}}{M} \quad (6)$$

Equations (5) and (6) are the same M-step updating equations as in GMM. The EM algorithm increases the log-likelihood consistently, while it will stop at a local maximum.

4.4 How Many Events?

Just like the magic number in clustering applications, the events number is also difficult to be determined. Fortunately, we can get the initial estimation of events number from the article count-time distribution. As shown in Figure 1, basically, each peak is corresponding to one event(in no overlapping situation), thus, our initial estimate of events number can be set as the number of peaks. However, since noises damage this distribution, there are too many peaks on plot of this distribution. We assume only the salient

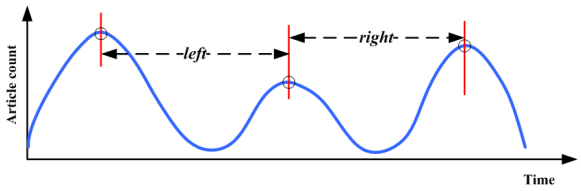

Figure 3: Salient score of the middle peak is the sum of left and right. left(right) is defined as the distance from current peak to the first higher peak on the left(right) hand

peaks are corresponding to events. To detect salient peaks, we define salient scores for peaks as:

$$score(peak) = left(peak) + right(peak) \quad (7)$$

Figure 3 illustrates the definitions of operator left and right of peak. The operators, ***left*** and ***right***, return distance to the most adjacent higher peaks.

In the initializing step, firstly, we use hill-climbing approach to detect all peaks, and then compute salient score for each of them. The top 20% peaks are defined as salient peaks, and the number of salient peaks is the initial estimation of k(number of events). 20% is an experimentally determined parameter. As an alternative way, user can specify the initial value of k(e.g. if user is only interested in the TOP 10 events, $k = 10$). Once we determine the initial estimation of k and the positions of salient peaks, we can initialize the events parameters correspondingly. Moreover, several different initial values of k can be got by splitting/merging initial salient peaks. Usually, peaks with lots of news articles or heavy tails may be mixtures of multiple events, thus, we split them to increase k and re-train models. To select the best events number, one available measure of the fitting goodness is the log-likelihood. Given this indicator, we apply the *Minimum Description Length*(MDL) principle [10] to select among values of k:

$$\begin{aligned} k &= \arg\max_k(\log(p(X;\theta)) - \tfrac{m_k}{2}\log(M)) \\ m_k &= 3k - 1 + k(N_p - 1) + k(N_l - 1) + k(N_n - 1) \end{aligned} \quad (8)$$

where the $log(p(X;\theta))$ is expressed in (1) and m_k is the number of free parameters needed for our model. As a consequence of this principle, when models with different values of k fit the data equally well, the simplest model is selected.

4.5 Event Summarization

In practice, we utilize two ways to summarize news events. On the one hand, we can choose some features with the maximum probabilities to represent event. For example, for event j, the 'protagonist' is the person with the maximum $p(person_p|e_i)$. Locations and keywords can be chosen similarly. However, the read abilities of such summarizations are bad. Thus, as an alternative way, we choose one news article as the representative for each news event.

Once we get the probabilistic distributions of persons, locations, keywords and time conditioned on events, we can assign news articles to events by Maximum a Posterior(MAP) principle:

$$y_i = \underset{j}{argmax}(p(e_j|x_i)) \quad (9)$$

where y_i is the label of news article x_i. The news article x_i

Multi-modal RED Algorithm:

1. Initializing events parameters
   a. Using hill-climbing algorithm to find all peaks
   b. Using salient scores to determine the TOP 20% peaks, and initialize events correspondingly
2. Learning model parameters
   a. E-step: computing posteriors by (3)
   b. M-step: updating parameters by (4), (5) and (6)
3. Increasing/decreasing the initial number of events until the minimum/maximum events number is reached
   a. Using Splitting/merging current big/small peaks, and re-initialize events correspondingly
   b. Goto step 2
4. Performing model selection by MDL as (8)
5. Summarizing

Figure 4: Summary of the proposed multi-modal RED algorithm

with the maximum $p(x_i|e_j)$ among articles assigned to the j-th event is a good representative of the event j, or the first article of each event is also a good representative.

4.6 Algorithm Summary

The whole process of the proposed multi-modal RED approach is summarized in Figure 4. The maximum and minimum numbers of events are determined experimentally.

5. APPLICATION: HISCOVERY SYSTEM

Based on the proposed event detection algorithm, we build a research system, HISCOVERY(HIStory disCOVERY), in which we provide two useful functions: Photo Story and Chronicle. In HISCOVERY, news articles come from 12 news sites, such as MSNBC, CNN and BBC. We run a web crawler once to get old news articles, and from then on, only trace the front pages of these sites to get the latest news articles.

5.1 Photo Story

Photo story is a rich representation of the past news even ts belonging to certain topic(e.g. "Halloween" is a topic,but each year's Halloween is an event). Usually, there are informative images embedding in news articles, which are very helpful to illustrate news events. By the proposed RED approach, news articles and their images are associated with found events. Figure 5 illustrates the user interface of Photo Story. Events and summaries are shown by their temporal order. We also use computer vision technologies to detect attention attracting areas (e.g. human faces), and then make a slides-show which emphasize on these areas.

5.2 Chronicle

Chronicles(e.g. chronicle of George W. Bush) provide very useful information, which are made manually by editors or history researchers nowadays. In HISCOVERY, the generation of a chronicle is constituted by three steps: i) user enters a topic, just like the query in Web search engine,

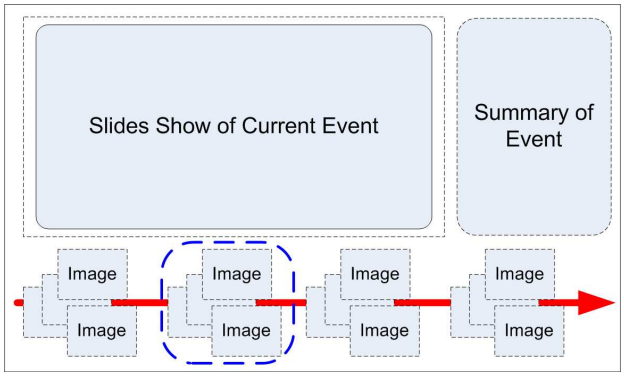

Figure 5: User interface of Photo Story. The bottom area shows events arranged in temporal order(each event is represented by a cluster of images), and the circled event is current event; the slide show of current event is provided at the top left area; and corresponding summary is presented at the top right area

Table 1: Details of dataset 1(part of TDT 4 dataset)

Time	Oct. 2000 - Jan. 2001
Number of articles	1923
Number of Topics(Events)	70
Average articles per event	27

ii) HISCOVERY searches our news corpus to gather related articles, and iii) the system utilizes the proposed RED approach to detect events belonging to this topic, and then sort summaries of events in chronological order. Since we have the images of the events, some representative images can also be shown in the final report.

6. EXPERIMENTAL METHODS

In this section, we explain the process of collecting experimental data, and then give the schemes of the empirical examinations.

6.1 Data Preparation

We prepare two datasets for experiments. The first is TDT [1] collections, which are benchmarks for event detection. We choose TDT4 dataset to run experiments, which contains 80 events annotated from almost 28,500 news articles. These articles were collected from the period of October 2000 to January 2001 from some sources like CNN, New York Times etc. Because the 80 events are only a part of events in the whole corpus, most articles are unlabeled. We extract those articles with labels as our dataset 1, and remove some events with few articles. More details of our dataset 1 are listed in Table 1.

For two reasons, we collect the second dataset: i) The period of TDT4 dataset is too short (only 4 months); ii) each topic consists of one event. That is, these events can be classified only by the contents, and the time model is not important. Here, topic is a bigger concept than event, which can contain many events. Thus, we choose three representative topics containing some events respectively. Table 2 illustrates the topics we chose. Once the topics are selected, we use the topic names as queries to search related news articles from some news websites, and then download the

Table 2: The details of our dataset 2

Topic	Time	#articles	#events
Earthquake	1995-2004	976	17
Halloween	1995-2004	762	9
Air Disaster	1995-2004	210	13

returned news articles. Three big news sites, CNN, MSNBC and BBC, are chosen as the sources of news articles. In the search results pages, news article's times are also returned, which can be extracted by simple wrappers.

Because extracting article's contents from HTML pages is a difficult problem, we use the remainder text by simply removing HTML tags and anchor text(according to our observation, in news articles, most irrelevant text are anchor text) as news article's contents. Because manually defining events are very subjective, we use similar methods to define and label events just like the TDT project. "Halloween" is a topic, which is reported once per year, thus, each year's reports can be regarded as an event; for "Earthquake" and "Air Disaster", their events lists can be found from corresponding official websites. We remove events without articles or with a few articles(less than 4 articles), and articles not belonging to any events.

We use the same way to process documents of the two datasets: extracting named entities, removing stop words and stemming. The named entities are extracted by BBN NLP tools [4], which can extract seven types of named entities, including persons, organizations, locations, date, time, money and percent, but we only use the first three kinds of entities(organization entities are merged into person entities).

6.2 Experimental Design

Three sets of experiments are performed in our study. The first experiment investigates the precision and recall of our approach on dataset 1. In this experiment, the reporting time of events are close or even overlapping, but their contents are different. Thus, the contents dominate the clustering results. In the second experiment, we examine the performance on articles distributing on long periods, where neither the contents nor time information are dominant. In the first two experiments, we set the cluster numbers as the true number of events, but in practice, the event number must be determined automatically. Thus, in the last experiment, we examine the performance under different events number(k), especially under the best k according to our model selection method.

To compare our approach with other algorithms, Yang et al.'s augmented Group Average Clustering(GAC) [12] and the generally used kNN algorithm [3] are chosen as baselines. Although, Yang et al.'s algorithm[12] were proposed several years ago, in terms of empirical results, it is still one of the best algorithms in TDT evaluations. In GAC, all articles are sorted in chronological order, and then an agglomerative clustering is performed. To accelerate this process, GAC split the corpus into fixed size buckets in its initializing step, and then clustering is performed within each bucket. To solve the problem that events may span on boundaries of buckets, a re-clustering strategy is utilized. Since there are some tunable parameters in GAC, we use the same settings as that in Yang et al. [12].

In the kNN algorithm, news articles are sorted in chronological order, and input into system one by one. For each incoming article, we compute its similarities with current cluster's centroids. If all similarities are less than a threshold, the coming article is labeled as the first article of a new event; otherwise it is merged into the most similar cluster, and the centroid of this cluster is updated. The value of the threshold is determined experimentally. This kNN algorithm is generally used in NED systems, and this algorithm is also used in some RED systems.

Table 3: A cluster-event contingency table

#	in event	not in event
in cluster	a	b
not in cluster	c	d

6.3 Evaluation Measures

TDT project has its own evaluation plan. However, their tasks are not consistent with ours. Thus, we choose the same evaluation metrics as that in Yang et al. [12]. Table 3 illustrates the two-by-two contingency table for a cluster-event pair, where a, b, c and d are document counts in the corresponding cases.

Three evaluational measures are defined using the contingency tables, including precision(p), recall(r) and $F1$ measure: Once got contingency tables and corresponding measures, we can evaluate global performance by averaging the three measures. The micro-average (by summing all contingency tables of all events, and then compute the three measures) and the macro-average(by averaging the three measures of all events) are generally used measures.

7. EXPERIMENTS AND RESULTS

We use C++ to implement all algorithms, and use Excel to plot all figures. The three compared approaches are termed as kNN, GAC and Probabilistic Model(ours), respectively.

7.1 Overall Performance on Dataset 1

This set of experiments compares the performances of the three approaches on dataset 1. In experiments, we set the events number as the true events number. Since GAC is a hierarchical clustering method, we stop after there are k clusters left, and run re-clustering 5 times as the recommended settings in Yang et al. [12]. For kNN, we report the results under the best threshold. Figure 6 illustrates the results of the three approaches.

Probabilistic model gains the best results, but the improvements are not significant. Because the dataset 1 contains articles published only in four months, corresponding events were reported closely or even overlapped. The contents of articles dominate the clustering processes; and the times are not very discriminative. Thus, although the GAC and kNN do not model times explicitly(only using chronological order), they also get good results. Modeling persons and locations by separate models brings extra computational costs to our approach, so we also run experiments to compare the performance with a simplified version of Probabilistic Model, in which the whole contents are modeled by one mixture of unigram model. Table 4 illustrates the results of the two approaches.

The better performance of the full Probabilistic Model indicates the benefits of modeling named entities by separate models. The named entities are very important for news

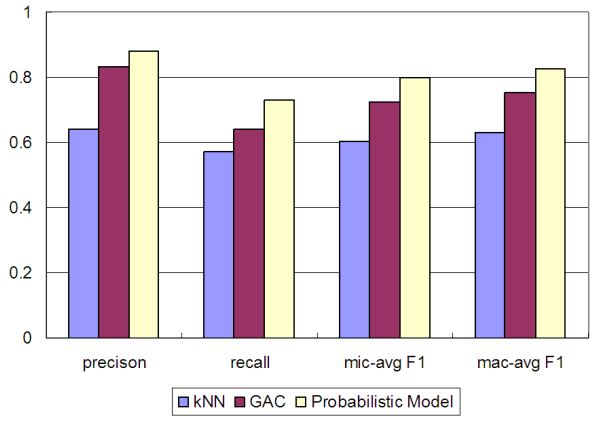

Figure 6: The experimental results of the three approaches on dataset 1

Table 4: Experimental results on dataset 1

	Probabilistic Model with NEs	Probabilistic Model without NEs
precision	0.847	0.743
recall	0.671	0.632
mic-avg. F1	0.749	0.683
mac-avg. F1	0.775	0.708

articles, but they are only a small part of contents. Modeling them separately can prevent them to be overwhelmed by tremendous other words. Although this strategy brings extra computational costs, the improvements are significant.

7.2 Overall Performance on Dataset 2

We run the three approaches on the three subsets of dataset 2 using the same settings as in experiment 1, respectively, and Figure 7 illustrates the results.

The results of Probabilistic Model are significantly better than GAC and kNN. By tracing the process of GAC, we find that in its initializing step, GAC split articles belonging to same events in different buckets. Because articles in dataset 2 are not uniformly distributed on time line, and GAC uses a fixed-size window (fixed quantity of articles) to split articles into buckets, this way causes splitting of some

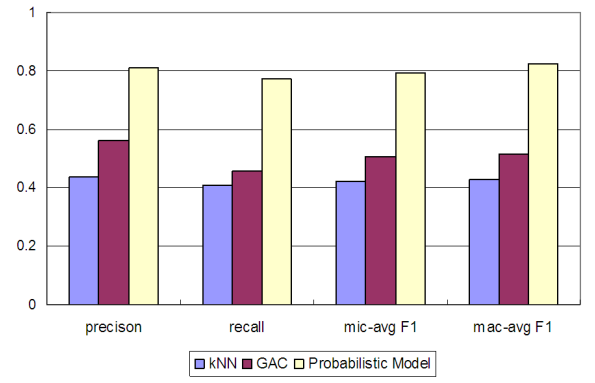

Figure 7: The experimental results of the three approaches on dataset 2

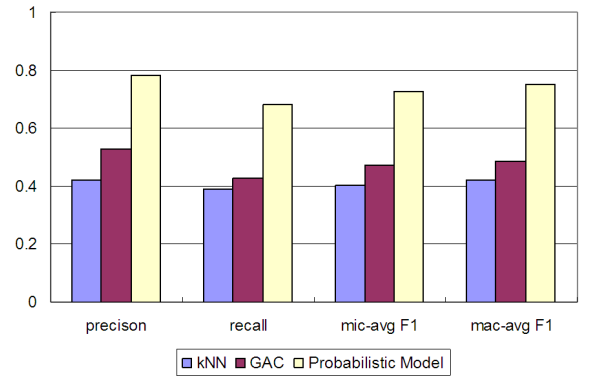

Figure 8: The results of three approaches on the new dataset, which is got by combining articles of the three topics in dataset 2

Table 5: Initial number of events got by salient peak analysis

	Earthquake	Halloween	Air Disaster
initial k	22	16	31

events. For kNN, the time information affects it little, but articles from different events may be assigned to the same cluster because it only considers the contents of news articles.

Furthermore, we combine articles of the three topics to get a bigger dataset, in which events belonging to the same topic are similar from the contents aspects, and at the same time many events are overlapping on time line. The classification boundary is determined by both the contents and time information. Figure 8 illustrates the results of three approaches on this dataset. Because the data becomes more confusing both from the contents aspects and the time aspects, the performances of GAC and kNN are worse than that in last experiment. However, the Probabilistic Model gets similar results in two experiments. This result indicates the effectiveness of modeling the time information explicitly.

7.3 How Many Events?

In the last two experiments, we set the number of events as the true number of events, thus, we examine the performance of model selection in this experiment. If we have no prior knowledge about the corpus and events, we can use the method proposed in section 4.4 to get an initial events number. In practice, we find the number of events is no more than 20% of the number of peaks of the articles count-time distribution. So we set the initial k to 20% of the number of peaks, and initialize all clusters according to these TOP 20% salient peaks. Table 5 illustrates the initial k's of the three topics in dataset 2.

We combine documents of the three topics to get a bigger corpus again, in which there are 39 events. Consequently, we initialize the number of events as the method proposed in section 4.4. There are totally 87 salient peaks. Then we increase and decrease the events number to compare the fitness of each approach. For each k, we obtain a partition of the dataset, and the events number is not equal to the true events number. Thus, instead of using the precision and recall measurements used in last two experiments, we

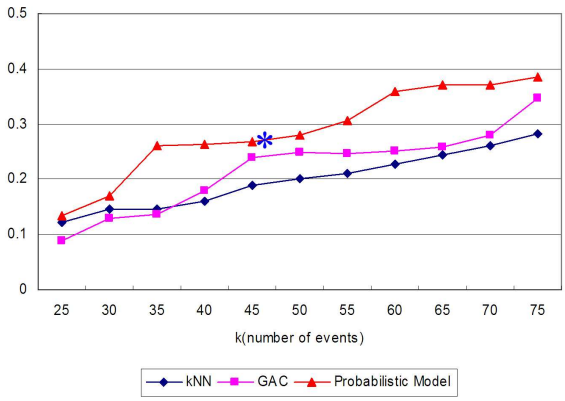

Figure 9: Mutual information of three approaches on the combined dataset. The position of the asterisk, 46, is the number of events selected with MDL principle

choose mutual information to measure the fitness of a partition with the ground truth, see also [9].

Figure 9 illustrates the experimental results of the three approaches on the combined data respectively. Under each event number, our approach fits the ground truth consistently better than the other two approaches. By model selection, the best events number we obtain on this combined dataset is 46, which is bigger than the true events number, 39.

8. CONCLUSTIONS AND FUTURE WORK

In this paper, we study the two characteristics of news articles and events, and accordingly propose a multi-modal RED algorithm. This algorithm is easy to understand and implement in practice. Both the contents and time information of news articles are modeled explicitly and effectively. Especially the model of timestamps works like auto-adaptive sliding windows on time line, which overcomes the inflexible usages of timestamps in traditional RED algorithms. Furthermore, the experimental results indicate the effectiveness of this approach.

In this paper, we do not touch how to find better representations of the contents of news articles (e.g. in [6] and [11]), i.e. feature selection. Next step, we will examine the performance of using approaches of [6] and [11] to refine the representation of the keywords vectors of news articles.

In practice, we also tried to use some other algorithms to model news events. Since news articles are typical time series data, it seemed that many dynamic models were fitful, such as *Hidden Markov Model*(HMM) and *Independent Components Analysis* (ICA). However, none of them get satisfied results in our experiments. Actually, our multi-modal RED approach is a simplified version of HMM. We think that noises and sparseness in feature space are main reasons which cause the failure of complex dynamic models. Although news articles are high quality parts of web documents, they yet can not be modeled easily. Thus, in future work, we will study how to deal with such issues, and use fitful dynamic models to model news events.

9. ACKNOWLEDGMENTS

Our thanks to those students took part in our tedious data labeling work. The insight comments from anonymous reviewers are greatly appreciated.

10. REFERENCES

[1] Topic detection and tracking(tdt) project. *homepage: http://www.nist.gov/speech/tests/tdt/.*

[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, R. Papka, and V. Lavrenko. On-line new event detection and tracking. *In Proc. of SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[4] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 1999.

[5] T. Brants, F. Chen, and A. Farahat. A system for new event detection. *In Proc. of the SIGIR conference on Research and development in information retrieval*, 2003.

[6] G. Kumaran and J. Allan. Text classification and named entities for new event detection. *In Proc. of the SIGIR Conference on Research and Development in Information Retrieval*, 2004.

[7] W. Lam, H. Meng, K. Wong, and J. Yen. Using contextual analysis for news event detection. *International Journal on Intelligent Systems*, 2001.

[8] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 2000.

[9] A. Strehl, J. Ghosh, and R. Mooney. Impact of the similarity measures on web-page clustering. *In Proc. of the AAAI 2000 Workshop on AI for Web Search*, 2000.

[10] J. F. Trevor Hastie, Robert Tibshirani. *The Elements of Statistical Learning Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer, 2001.

[11] Y. Yang and J. Z. et al. Topic-conditioned novelty detection. *In Proc. of the SIGKDD international conference on Knowledge discovery and data mining*, 2002.

[12] Y. Yang, T. Pierce, and J. G. Carbonell. A study on retrospective and on-line event detection. *In Proc. of the SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[13] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. *In Proc. of the SIGIR Conference on Research and Development in Information Retrieval*, 2002.