*2006 IEEE International Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan*

Anticipatory Event Detection via Sentence Classification

Qi He, Kuiyu Chang, Ee-Peng Lim

Abstract— **The idea of event detection is to identify interesting patterns from a constant stream of incoming news documents. Previous research in event detection has largely focused on identifying the first event or tracking subsequent events belonging to a set of pre-assigned topics such as earthquakes, airline disasters, etc. In this paper, we propose a new problem, called Anticipatory Event Detection (AED), which aims to detect if a user-specified event has transpired. AED can be viewed as a personalized combination of event tracking and new event detection. We propose using a sentence classification approach to solve the AED problem for a restricted domain; detecting articles that describe final game results from NBA basketball news. Experimental results demonstrate the feasibility of our proposed AED solution.**

I. INTRODUCTION

With the advent of mobile Internet-enabled devices such as GPRS and 3G equipped mobile phones, the Internet has gradually evolved into the next real-time universal information source. Professionals from specialized industries such as the financial sector, and who rely on real-time information for decision-making, are already using SMS (Short Messaging System) news alerts delivered directly to their mobile phones 24/7. Such a news alert prototype has been previously reported in [10].

In general, subscribers do not wish to be awaken in the middle of the night by interesting but irrelevant events. Ideally, a news alert system should acquire the preferences of a subscriber over time, so that the system only sends relevant alerts. In practice, the subscriber will have to specify the kind of events he/she is interested in, e.g. by supplying a phrase like "Ming YAO wins basketball match".

This paper is motivated by one particular aspect of SMS news alerts, i.e. how to identify sentences and therefore documents signifying that a user-defined/anticipated event has actually occurred? We call this new problem AED (Anticipatory Event Detection). AED relies on a DET (Defined Event Trigger), which is comprised of a set of user-supplied keywords and trained ET (Event Transition) models.

AED thus boils down to classifying sentences into those that consume a predefined event (hit) and those that do not (miss). However, sentence retrieval is a very difficult problem [8] by itself, largely due to the sparsity of features (few words in a sentence), and the lack of overall language context. Fortunately, we found that by enhancing the context semantics (by extracting the named entities) of each sentence, decent classification accuracy can be achieved using a simple bag-of-words approach.

This paper thus investigates various sentence retrieval strategies for AED, with a substantial focus on improving retrieval quality. The remaining of this paper is organized as follows. Section 2 surveys related work. In Section 3, we present the problem definition for AED, compare it to existing event detection tasks, and subsequently propose two sentence classification approaches. Section 4 elaborates on our two proposed sentence classification models for AED. Section 5 describes our AED dataset, which is restricted to the domain of identifying anticipatory events concluding the win/loss of a basketball match. Section 6 describes our experimental setup. Section 7 summarizes our AED results, and section 8 concludes the paper with a discussion of future work.

II. RELATED WORK

AED can be viewed as a special case of the more general research area collectively known as Topic Detection and Tracking (TDT). In fact, New Event Detection (NED) and Topic Tracking (TT) in news stories [5], [6], [7], [9], [13], [18] from TDT research comprise a significant body of related work. TDT addresses event-based organization of news stream [5]. Within TDT, a topic is defined as an event or activity, along with all directly related events and activities [3]. However, AED differs from classical TDT mainly in the aspect that it will only return a user specified event that has fired.

There is a concept of binary state in AED; the anticipated event can either be consumed or not. In general, NED will detect and return all new events of a particular topic, and TT will detect and return all new developments of a topic, whereas AED will detect and return the document that has fired based on a user specified binary transition. For example, on the topic of earthquakes, NED will detect the first story about *any* earthquake. TT will detect *any* new developments pertaining to a specific earthquake. In contrast, AED will fire only when a state specified by the user has been reached. For example, the firing state could be "Earthquake strikes major Chinese city with heavy casualties". As such, AED seemingly combines elements from both NED and TT. A more detailed comparison between AED and TDT is given in Section 3.

In the area of machine understanding of events, Nallapati et al. [17] built a cascading structure for events belonging to

Qi He is with the Division of Information Systems, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798. qihe@pmail.ntu.edu.sg

Kuiyu Chang is with the Division of Information Systems, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798. kuiyu.chang@pmail.ntu.edu.sg

Ee-Peng Lim is with the Division of Information Systems, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798. aseplim@ntu.edu.sg

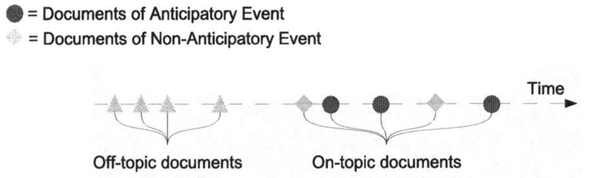
1-4244-0100-3/06/$20.00 ©2006 IEEE

one topic, based on the belief that hierarchical models are more effective than flat structures in capturing semantics of on-topic stories. Other work that also looked at the structure within a topic are [19], [15]. Our AED solution is influenced by these ideas but we model the structure of a topic manually by classifying sentences representing the "before" and "after" states of an anticipatory event. Our basic assumption is that one successfully classified sentence is enough to indicate a hit ("after") document in AED.

III. Anticipatory Event Detection (AED)

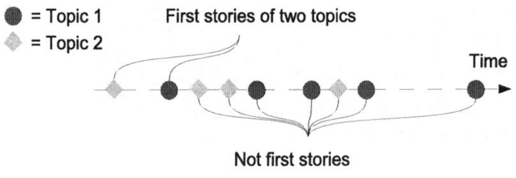
A. Problem Definition

New events occur every day. It is impractical to bombard users with every new event. Instead, we would like to alert a user only to events that interest him. Thus, the objective is to discover and monitor evolutions of a predefined news event such as the capture of Osama Bin Laden, the result of the US Election, the updated local death toll from the 2004 Tsunami, the outcome of a single NBA basketball match, etc. We define this problem as Anticipatory Event Detection (AED), distinguishing it from other related problems like Topic Tracking (TT) and New Event Detection (NED). Anticipatory Event Detection (AED) *detects transpired events based on some user preferences* [12]. The idea is to alert the user to a specific anticipated event, e.g. events involving a major shift/change in information content.

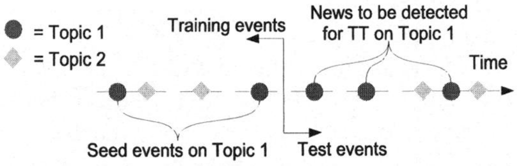


● = Documents of Anticipatory Event
◆ = Documents of Non-Anticipatory Event
Time
Off-topic documents
On-topic documents

Fig. 1. Anticipatory Event Detection model. A document may be off-topic or on-topic, with some on-topic documents describing related (e.g. to the current game) but historical (e.g. previous game scores) events.



(a) New Event Detection (NED).



(b) Topic Tracking (TT).

Fig. 2. TDT models defined by TREC [4].

Figures 1 and 2(a) show the AED and NED models for news events, respectively. While AED returns only fired/transitioned anticipatory events, NED returns the first document for *all* new events. The TT model in Figure 2(b) is mainly concerned with classifying documents into various topics, so that they can be tracked over time. Moreover, instead of examining the semantics of a tracked event, the vast majority of TT research simply uses the document similarity approach to determine if an incoming document should be tracked.

B. Proposed Approach

One way to look at AED is to think of it as finding the transition between two adjacent events in an *event transition graph* whose events are represented by news articles covering the event transition graph before and after a particular transition has consummated. Figure 3 shows an *event transition graph* with n events and $n-1$ transitions for $topic_i$ (e.g. eBay acquires Skype). A user may only be interested in receiving a notification when a *particular* transition has fired, and not be bothered about the other transitions. If sufficient number of news articles can be collected for each of the events, it would be possible to detect any of the $n-1$ transitions. In order to learn a particular transition, a model will have to be trained to classify articles as occurring "before" or "after" the transition. For example, given $transition_{1,2}$ in the *event transition graph* of Figure 3, a user defined AED could be to detect the first story of $event_2$.

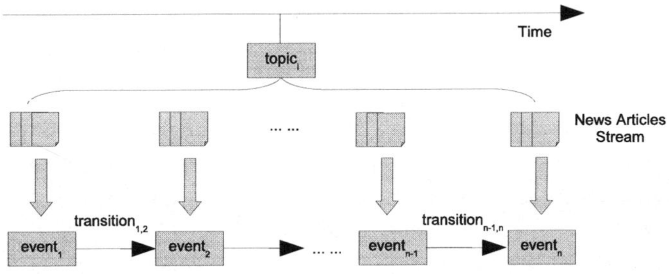


Fig. 3. Anticipatory Event transition graph.

Other than topic classifiers, NED and TT typically require no training. On the other hand, AED deals with the transition from the "pre" state to the "post" state of a user defined event. For arbitrary transitions, this is an open-ended problem as there exists no simple machine-interpretable syntax to define one.

Thus, our approach is to first select a specific topic of interest and identify a transition. Next, we collect a set of training documents containing a mix of "pre" and "post" documents. Then, each sentence in a training document is manually labelled as either negative ("pre") or positive ("post"). Lastly, a sentence classifier is trained from all labelled sentences.

In general, a fired event transition can be confirmed from a sentence, given enough context. For example, the following sentence would qualify as a "hit" sentence for the anticipatory event "win basketball match".

"The Knicks outscored Philadelphia 32-22 in the fourth quarter to secure the win."

To identify the event context, we introduce the Defined Event Trigger (DET), which comprises user preferences defined by a set of keywords and the event transition model learnt from the training data. In the above example, the user preference is defined as "win basketball match" and the event transition model could simply be a sentence classifier. So far, traditional document retrieval techniques have proven unsuccessful when applied to sentence retrieval [8], even with richly enhanced queries derived from TREC topic descriptions. This is because a single sentence contains neither enough information (curse-of-dimensionality) nor context to form a meaningful model.

IV. Sentence Classification Models for AED

Based on the observation that the most representative sentences in a positive document typically provide a good summary of the event transition, we propose two approaches to solve the AED problem, namely a flat and two-level hierarchical SVM sentence classifier. The sentence classification models were part of a pilot study on the AED problem[1].

The single-level SVM sentence classifier classifies all sentences as positive (on-topic and event confirming) or negative (off-topic, or on-topic but non-event confirming). The two-level SVM classifier attempts to distinguish sentences describing current events from those dealing with historical[2] events.

The two-level approach overcomes the problem of a single-level topic classifier failing to distinguish between current ("positive") and historical on-topic sentences ("historical"), both of which share a common vocabulary. A two-level approach is preferred over a 3-class model which would otherwise suffer from too much feature cross-talk between the historical and positive class.

For example, "*the rejuvenated Celtics have won three straight since then and six straight at home overall*" is a typical historical sentence that is considered on-topic by a single-level classifier. This confusion is easily resolved by using a second level classifier trained specifically to discriminate the current on-topic events from historical ones.

Figure 4 shows the structure of the two-level SVM classifier. The first level classifier aims to detect all on-topic sentences, which include both positive and historical sentences. The second level classifier subsequently classify the on-topic sentences into positive or historical.

V. Anticipatory Event Dataset

In order to evaluate the performance of our proposed AED solution, we picked basketball matches as the topic of interest. Since AED is a new area defined by us, with no standard evaluation benchmark dataset, we created our own dataset, *basket100*, which is based on the user preference "win basketball match". The *basket100* collection comprises 100 documents returned by Google using the search terms "win basketball match".

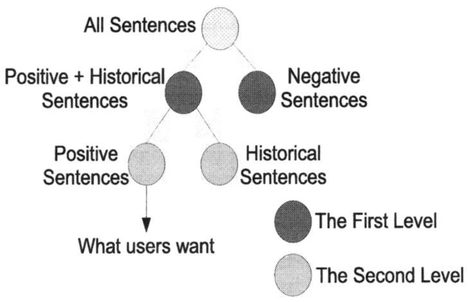


Fig. 4. Two-level SVM sentence classifier.

A. Distribution

In *basket100*, 93 out of 100 documents are relevant, i.e. describes basketball games, and the remaining 7 are irrelevant. The collection contains 2340 sentences, comprising 4499 unique terms (words). The 2340 sentences were manually annotated into 3 categories:

1) positive-current class for "current basketball result"
2) negative-historical class for "historical basketball results"
3) negative class for "irrelevant" or off-topic sentences

Table I shows the summary statistics for *basket100.*

TABLE I
Class distribution of basket100.

Classes	*Count*
Positive documents (class 1: win basketball event)	93
Negative documents (class 2: irrelevant)	7
Total	100
Positive sentences (class 1: current win basketball event)	189
Negative sentences (class 2: historical win basketball event)	117
Negative sentences (class 3: other irrelevant sentences)	2034
Total	2340

B. Sparsity

Sparsity, which computes the fraction of unique terms in each sample, is a very important factor affecting sentence retrieval accuracy, and is shown in Table II. The sparsity

TABLE II
Sparsity statistics for basket100.

	document model	*sentence model*
Average term count per sample	568.6	24.3
Average unique term count per sample	185.7	12.5
Total unique term count	4499	4499
Sparsity	95.87%	99.72%

[1] An improved AED model is described in a subsequent paper [12].

[2] There are only minor semantical differences between historical and current events in the limited context of a sentence. In practice, historical events are not newsworthy and are therefore discarded.

metric is defined as:

$$Sparsity = \frac{T - A}{T} \quad (1)$$

where T is the total unique term count and A is the average unique term count per sample.

From Tables I and II, we can see that our testbed is unbalanced and very sparse, which is rather typical of real world text data. Note that for each document or sentence, the average number of unique terms is much smaller than the average number of words, due to multiple occurrences of popular terms.

VI. Experiment Setup

A bag-of-words approach based on variations of Term and Document Frequency (such as TFIDF) and enhanced with named entities, were used to represent each sentence. The software package SVM-light [2] was used to build the AED sentence classifiers.

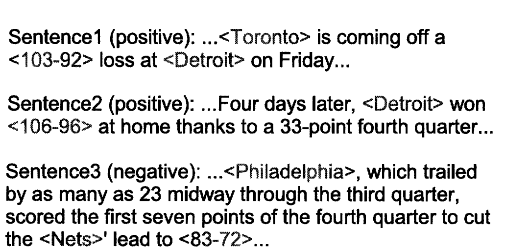
A. Preprocessing

Version 1.4.3 of the open source Lucene[1] software was used to tokenize the news text content, remove stop words, and generate the document-word vector. In order to preserve time-sensitive past/present/future tenses of verbs, no stemming was done other than the removal of a few articles. The dataset was divided into training and test partitions. SVM Cost factors[16] were used to deal with the highly unbalanced class sizes.

B. Named Entities

Named Entities were originally created for extracting information from unstructured text [11], which primarily includes extracting names of people, places, organizations, etc. Named entities are thus extremely useful for text understanding by allowing us to answer "wh-" type of questions such as "who", "where", and "what" of a specific event. Kumaran [14] applied named entities to NED and achieved varying degrees of success on different categories of events.

For sentence retrieval, named entities are especially helpful, as they greatly enhance the limited context. We manually extracted the game scores and basketball team names as two types of named entities. Some examples of named entities are marked-up with "<>" in Figure 5.



Sentence1 (positive): ...<Toronto> is coming off a <103-92> loss at <Detroit> on Friday...

Sentence2 (positive): ...Four days later, <Detroit> won <106-96> at home thanks to a 33-point fourth quarter...

Sentence3 (negative): ...<Philadelphia>, which trailed by as many as 23 midway through the third quarter, scored the first seven points of the fourth quarter to cut the <Nets>' lead to <83-72>...

Fig. 5. Some extracted named entities from the *basket100* dataset.

We observed most positive sentences to contain at least one team name and one score. However, some historical sentences also contain a team name and score. For example, sentence3 in Figure 5 contains two team names and one score, but is in fact a typical negative sentence reporting mid-game scores (no conclusive win). Clearly, isolated named entities may be used to enhance the feature vector of sentences as in our approach, but the presence of which alone cannot be used to detect an anticipatory event.

C. Evaluation Methodology

Our experiment results are evaluated using the standard precision and recall measures, defined as:

$$Precision = \frac{\#\ correct\ positive\ predictions}{\#\ positive\ predictions} \quad (2)$$

$$Recall = \frac{\#\ correct\ positive\ predictions}{\#\ positive\ samples} \quad (3)$$

To avoid "overfitting" the training data, we use 10-fold cross validation to compute the average test precision and recall for all experiments. The harmonic mean of the average precision and recall (F1-Score or F-Measure), computed as shown below, provides the overall AED accuracy.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

D. Methods to be Evaluated

In practice, the term weighting scheme used to represent a sentence vector has a significant impact on the classification accuracy. The following methods using different term weighting schemes were compared in our experiments:

- Single-level Classifier with {Standard TF, TFIDF, TF-ISF, TF+named entity features}
- Two-level Classifier with {Standard TF, TFIDF, TFISF, TF+named entity features}

The standard TF scheme simply uses the raw frequency count of each term within a sentence. Another important factor to consider is the distribution of terms across a collection. Usually terms that are limited to a few sentences are useful for discriminating those sentences from the rest of the collection. This assumption leads to the *inverse sentence frequency* (ISF) weighting scheme. Likewise, the classical *inverse document frequency* (IDF) was also applied to each sentence assuming that terms appearing in a small number of documents are more useful. The various term weighting schemes are summarized as follows:

Standard TF	:	tf_{ij}
TFIDF	:	$tf_{ij} \times log(N_d/n_i)$
TFISF	:	$tf_{ij} \times log(N_s/s_i)$

where tf_{ij} is the frequency of the i-th term in the j-th sentence, N_d is the total number of documents in the collection, N_s is the total number of sentences in the collection, n_i is the number of documents containing the i-th term, and s_i is the number of sentences containing the i-th term. Our proposed weighting scheme, TF with named entities, simply appends two additional dimensions representing the extracted team name and game score to the TF sentence vector.

VII. Experiment Results

The two classifiers using various term weighting schemes were applied to the *basket100* dataset.

A. Single-level SVM Sentence Classifier

Figure 6 shows the classification results of the single-level SVM using various term weighting schemes. We see that the sentence classifier using our proposed weighting scheme yielded the best F-Measure of 0.69, leading the next compeititor by 15%. Moreover, the recall of 0.63 is low by practical standards, despite it beating the nearest competitor (TF) by more than 20%.

The other methods fared significantly worse. TFISF performed worse than TF, probably due to the fact that there were too many negative (including historical) sentences, thereby distorting the ISF. Note that for single-level classification, the positive and historical winnings are labelled differently, despite them sharing a common vocabulary, e.g. “win”, ”loss”, etc. TFIDF performed the worst, due to the large discrepancies between the importance of a term at the sentence and document level.

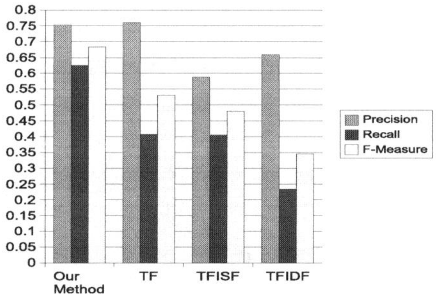


Fig. 6. Cross validated (10-fold) results of single-level SVM classifier.

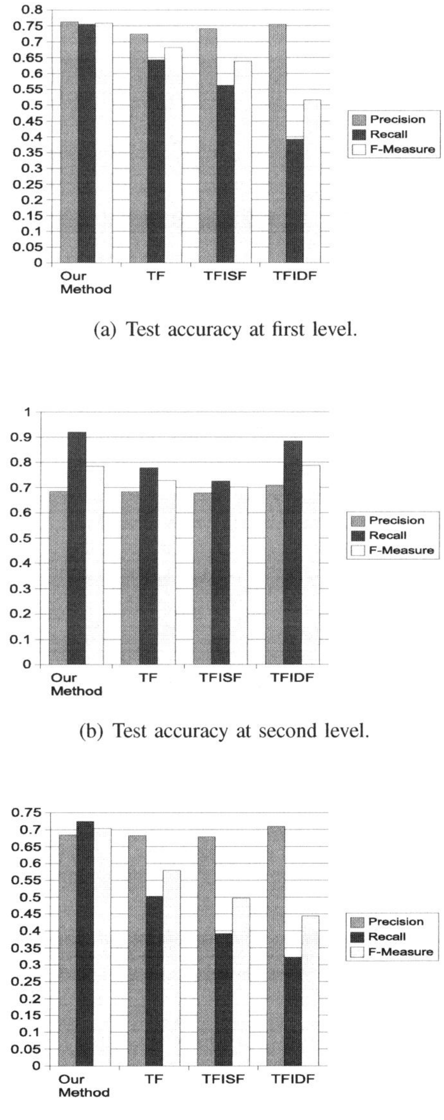
B. Two-level SVM Sentence Classifier

Figure 7 shows the results of the two-level classifier using different term weighting schemes. Since the first level classifier is only responsible for distinguishing on-topic sentences from off-topic ones, its performance was measured based on all on-topic sentences which included historical sentences.

Figures 7(a)-(b) show that the precision values at both levels were not affected much by the different weighting schemes, unlike with the single-level classifier. This confirmed our previous suspicion that the similarity between positive and historical sentences was a large contributing factor to the low precision when inverse document and sentence frequencies come into play for the single-level classifier. The overall performances of the two-level classifier is shown in Figure 7(c), with our method achieving the overall best result of 0.69 precision and 0.72 recall.

C. Discussions

Since the second level classifier is trained to differentiate only between positive and historical on-topic sentences, it is completely clueless about any false positive (off-topic/misclassified) sentences trickling down from the first level. In theory, the second level classifier will randomly classify these sentences, i.e. half of the false positive samples from level one will be classified into positive and the other half into historical. We have verified this indeed to be true in our experiments. Clearly, the first level classifier is crucial to the success of the second level classifier, and thus the overall accuracy.



(a) Test accuracy at first level.

(b) Test accuracy at second level.

(c) Overall test accuracy.

Fig. 7. Cross validated (10-fold) results of two-level SVM classifier.

Figures 6 and 7(c) also shows the F-Measure results for the single and two level classifers using various term weighting schemes. Apparently, a two-level sentence classifier based on our named entity enhanced TF weighting yields the best overall performance in terms of F-Measure.

Numbers aside, the practical implications for AED is simply as follows. If high precision is desired, go with the single-level classifier. This means that if a subscriber awakens at night, it is probably due to a valid news event. However, he may miss out on some important events due to the low recall. On the other hand, if he is willing to put up with 6% less precision (i.e. awaken up by more irrelevant alerts), he stands to catch 10% more (recall) of the actual alerts by using the 2-level classifier. Thus, each approach has its pros and cons, and the ultimate choice is best left to the news alert subscriber.

VIII. Conclusion and Future Work

We proposed a new practical application called Anticipatory Event Detection (AED), which is a more refined and personalized form of event tracking and detection. We then investigated two sentence-classification based methods to tackle the AED problem, which were verified experimentally on a restricted domain. In the process, we discovered that sentence classification accuracy is affected significantly by the choice of weighting schemes, and that our approach of using TF with named entities provided the best results. We also found that by incorporating more semantical structure into the classifier model, i.e. by using two levels, the overall performance was improved slightly, with significantly higher recall at the expense of reduced precision.

Sentence classification is not new, but up till now, results previously reported for sentence classification have been very dismal [8]. Thus, another contribution of our paper is to demonstrate that good sentence classification performance (around 70% precision and recall) is attainable if the problem domain is well-defined and restricted, such as for AED. Nevertheless, we showed in another paper [12] that a document based classification approach achieved much better results as the document contains significantly more information.

The main limitation of AED lies in its reliance on a pretrained transition model for every user-specified anticipatory event. This means that in practice, a user is not allowed to specify any anticipatory event, but instead must choose from a small list of available pre-trained anticipatory events, e.g. mergers and acquisitions, sports scores, etc. This is acceptable if we can train a list of AED models satisfying 80% of the users. Ideally, future work in AED should study ways to allow users to define arbitrary anticipatory events, from which training data can be semi-automatically collected from the Internet to generate the appropriate transition models. Moreover, each of these steps represent a major development milestone in natural language understanding.

For the foreseeable future, we envisage a real-time feedback AED system that allows a subscriber to refine his/her anticipatory event definition using similar historical events. For example, to define an anticipatory event such as “China attacks Taiwan”, the user can specify a similar transpired event like “Iraq invades Kuwait”, and manually supply the set of “pre” and “post” documents of the historical event, from which the AED system can learn the transition. With the above improvements, the AED system could very well become a truly reliable and personalized news alert system that people can put to good practical use.

References

[1] *Apache lucene 1.4.3, http://lucene.apache.org.*

[2] *Svm-light, http://svmlight.joachims.org/.*

[3] *Tdt: Annotation manual version 1.2, august 4 2004, http://www.ldc.upenn.edu/projects/tdt2004.*

[4] *Topic detection and tracking research, http://www.nist.gov/speech/tests/tdt/index.htm.*

[5] James Allan, *Topic detection and tracking. event-based information organization*, Kluwer Academic Publishers, 2002.

[6] James Allan, Hubert Jin, Martin Rajman, Charles Wayne, Daniel Gildea, Victor Lavrenko, Rose Hoberman, and David Caputo., *Topic-based novelty detection: Final report*, In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, 1999.

[7] James Allan, Victor Lavrenko, and Hubert Jin, *First story detection in tdt is hard*, In Proceedings of the 9th ACM conference on Information and knowledge management (CIKM), 2000, pp. 374–381.

[8] James Allan, Courtney Wade, and Alvaro Bolivar, *Retrieval and novelty detection at the sentence level*, In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 2003, pp. 314–321.

[9] Throsten Brants, Francine Chen, and Ayman Farahat, *A system for new event detection*, In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 2003, pp. 330–337.

[10] Kaixin Chua, Wee-Shiang Ong, Qi He, Kuiyu Chang, and Albert Kek, *Intelligent portal for event-triggered sms alerts*, In Proceedings of the 2005 IEE Mobility Conference, 2005.

[11] Ralph Grishman and Beth Sundheim, *Message understanding conference - 6: A brief history*, In Proceedings of the 16th International Conference on Computational Linguistics, 1996, pp. 466–471.

[12] Qi He, Kuiyu Chang, and Ee-Peng Lim, *A model for anticipatory event detection*, In Proceedings of the 25th International Conference on Conceptual Modeling (ER), 2006.

[13] Hubert Jin, Rich Schwartz, Sreenivasa Sista, and Frederick Walls, *Topic tracking for radio, tv broadcast, and newswire*, In Proceedings of the DARPA Broadcast News Workshop, 1999, pp. 199–204.

[14] Giridhar Kumaran and James Allan, *Text classification and named entities for new event detection*, In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004, pp. 297–304.

[15] Dawn Lawrie and W. Bruce Croft, *Discovering and comparing topic hierarchies*, In Proceedings of the RIAO 2000 Conference, 1999, pp. 314–330.

[16] Katharina Morik, Peter Brockhausen, and Thorsten Joachimss, *Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring*, In Proceedings of the 16th International Conference on Machine Learning (ICML), 1999, pp. 268-277.

[17] Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan, *Event threading within news topics*, In Proceedings of the 13th International Conference on Information Knowledge Management (CIKM), 2004, pp. 446–453.

[18] Nicola Stokes and Joe Carthy, *Combining semantic and syntactic document classifiers to improve first story detection*, In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001, pp. 424–425.

[19] Aixin Sun and Ee-Peng Lim, *Hierarchical text classification and evaluation*, In Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM), 2001, p. 521C528.