2007

Anticipatory event detection via classification

He QI

Kuiyu CHANG
Nanyang Technological University

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

DOI: https://doi.org/10.1007/s10257-007-0047-z

ISeB (2007) 5:275–294
DOI 10.1007/s10257-007-0047-z

ORIGINAL ARTICLE

Anticipatory event detection via classification

Qi He · Kuiyu Chang · Ee-Peng Lim

Published online: 1 March 2007

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 describe 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 sentence-level and document-level classification approaches to solve the AED problem for some restricted domains; given some user preferred topic event transition, we first train the corresponding event transition model, and then detect the occurrence of the transition for the stream of news covering the topic. Our experimental results on both terrorist-related and commercial events demonstrate the feasibility of our proposed AED solutions.

1 Introduction

Open Source Intelligence (OSI) plays a fundamental role in Intelligence and Security Informatics (ISI), accounting for as much as 80% of the overall intelligence (Quiggin 2006). In fact, former US Joint Chiefs Chairman and former Secretary of State Colin Powell said: “I preferred the Early Bird with its compendium of newspaper stories to the President’s Daily

Q. He · K. Chang (✉) · E.-P. Lim
School of Computer Engineering, Nanyang Technological University,
Singapore 639798, Singapore
e-mail: kuiyu.chang@pmail.ntu.edu.sg

Brief, the CIA's capstone daily product''. Thus, the ability to constantly monitor and accurately track events from news sources all over the world is vital to ISI.

Major online portals like Google and Yahoo allow users to subscribe to news alerts by specifying a list of present/absent keywords to define a particular event that he or she is interested in. Unfortunately, current alert systems are not smart enough to figure out whether a news document containing all the user defined words actually confirms occurrence of the the event. In fact, some service providers like Yahoo still entrust a human operator to approve system triggered news alerts, whereas others like Google prefer to use a completely automated approach, at the expense of generating many false alarms/alerts (He et al. 2006b).

New events occur every day. In general, subscribers do not wish to be awaken in the middle of the night by interesting but irrelevant events. Moreover, it is impractical to bombard users with every new event. Instead, we would like to alert the users to events that they are interested in. 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 news alerts, i.e., how to identify sentences and documents signifying that a user-defined/anticipated event has actually occurred. Thus, we wish to discover and monitor transition of a predefined news event, such as the capture of Osama bin Laden, the result of the US Election, the death toll from the 2006 Java earthquake, the outcome of a single NBA basketball match, etc. We call this new problem anticipatory event detection (AED).

One way to look at AED is to think of it as finding the transition between two adjacent events in an event transition graph (ETG) whose events are represented by news articles covering the event transition graph before and after a particular transition has consummated. A user may only be interested in receiving a notification when a particular transition has fired, and not be bothered about the remaining transitions. If sufficient number of news articles can be collected for each of the events, it would be possible to detect any transition. In order to learn a particular transition, a model will have to be trained to classify articles as occurring ''before'' or ''after'' the transition.

Anticipatory event detection thus boils down to classifying documents into those that consume a predefined event (hit) and those that do not. The classification of sentence for AED is also studied for the recent booming development in wireless information (Strader et al. 2004) with the advent of mobile Internet-enabled devices such as GPRS and 3G equipped mobile phones. Such a news alert prototype via mobile communication has been previously reported by Chua et al. (2005). Our preliminary results on AED have been quite encouraging, thus paving the way for further investigation into this area. Our contributions are summarized as follows:

1. Proposed AED as a new research area with real-world relevance to the online tracking of ISI events.
2. Defined a sentence model in our pilot AED study (He et al. 2006a).
3. Followed up with a document AED model (He et al. 2006b).
4. Proposed different ways of applying named entities in the sentence and document classification models.
5. Verified the feasibility of AED in practice for two restricted domains, NBA basketball matches, and Mergers and Acquisitions.

The rest of this paper is organized as follows. Section 2 surveys related work. In Sect. 3, we present the problem definition for AED, compare it to existing event detection tasks, and subsequently propose the framework of AED. Section 4 elaborates on our sentence classification model for AED and Sect. 5 describes our document classification model. In Sect. 6, experimental data are introduced. Section 7 provides a discussion of the AED experimental results, and Sect. 8 concludes the paper with a discussion of future work.

2 Related work

Anticipatory event detection can be viewed as a special case of the more general research area collectively known as topic detection and tracking (TDT) (Topic detection and tracking research, http://www.nist.gov/speech/tests/tdt/index.htm). In fact, new event detection (NED) and topic tracking (TT) of news stories (Allan et al. 1999, 2000; Allan 2002; Brants et al. 2003; Jin et al. 1999; Stokes and Carthy 2001) from TDT research comprise a significant body of related work. TDT addresses event-based organization of news stream (Allan 2002). Within TDT, a topic is defined as an event or activity, along with all directly related events and activities (Tdt: Annotation manual version 1.2, august 4 2004, http://www.ldc.upenn.edu/projects/tdt2004). 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 consummated 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.

In the area of machine understanding of events, Nallapati et al. (2004) built a cascading structure for events belonging to 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 includes (Lawrie and Bruce Croft 1999; Sun and Lim 2001). Our AED solution is influenced by these ideas but we model the structure of a topic by classifying sentences/documents representing the ‘‘before’’ and ‘‘after’’ states of an anticipatory event. Our basic assumption is that one successfully classified sentence/document is enough to indicate the occurrence of the AE.

A generalized hierarchical binary decomposition of output space framework was previously proposed by Kumar and Ghosh (1999). Chen et al. (2004) later proposed a more sophisticated hierarchical SVM (HSVM) framework by recursively solving the max-cut problem at each level. They showed that the binary HSVM classifier outperformed other flat classifiers. In this paper, we employ a simple one-class versus others multi-class support vector machine (SVM) (Cortes and Vapnik 1995) classifier and a two-level binary SVM classifier. The two-level binary SVM classifier simply cascades two binary SVM classifiers, with each classifier trained independently of the other.

3 Anticipatory event detection

The original motivation for AED is to let users receive only anticipated news alerts, i.e., those alerts that interest the user and which usually involve a major shift/change in information content, thereby saving users from having to deal with unnecessary interruptions. A formal definition is given as follows:

Defintion 1 (**AED**) AED monitors news streams to detect confirmed user-anticipated events based on some user preferences.

A user preference is defined as a user selected anticipatory event transition (AET) among several different event transitions for a given topic type. Figure 1 shows a global time-ordered sequence of documents, some on-topic and others off-topic with respect to an AET. Among the on-topic documents, only those that confirms the AET are considered hit documents, and should be identified by an AED system.

Actually, topics belonging to the same type (e.g., election of US President, etc.) often involve a common set of event transitions (e.g., nomination of party’s Presidential candidates, nomination of party’s Vice-Presidential can-

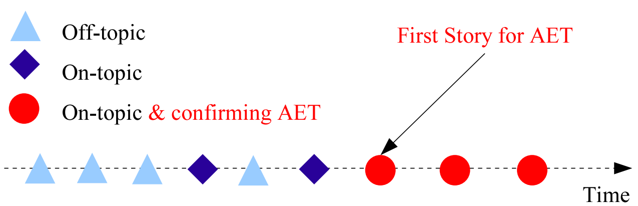


Fig. 1 Anticipatory event detection (AED) model. Among the on-topic documents, only those that confirm the consummation of the AE are identified

didates, election of party's Presidential team, election of Presidential team, etc.). We propose to model the multiple event transitions of the same topic type by an event transition graph. A user can select one of the several available event transitions as the desired anticipatory event transition, which means that the system should detect the first news after the transition.

As illustrated in Fig. 2, we have an example event transition graph describing a common structure shared by all company acquisition topics. Suppose a user is interested in the event transition, $t_{2,3}$ from event e_2 (in talks to acquire) to event e_3 (announce acquisition). The figure shows that there can be multiple news articles associated with the ''Announce acquisition'' event and the first one among them will be first story triggering the transition and is therefore the target of detection.

Our AED classification framework is given in Fig. 3. We assume the existence of a relevant event document set that can be used to train a classifier to learn the event transition model. Once trained, the model is able to detect news articles belonging to any desired event in the event transition graph, which is typically the destination event of the user preferred (selected) transition. The learnt model can then be applied to identify the first story after the desired AE transition.

To train the AED system for a given transition, we first retrieve a set of *generic* news articles from Google News Alerts based on the user supplied list of domain specific keywords (excluding any named entities). The articles or sentences are then manually labelled as positive or negative (with respect to the user selected transition), and fed into a classifier for training. For a test topic, we use the trained generic model to detect the earliest news article published after the AE has consummated.

4 Sentence classification model for AED

4.1 Motivation

Humans can easily decide if a news document correspond to a hit (transition fired) or not just by looking at the title and first few sentences. Based on this intuition, one or more key sentences in a positive document should provide enough information to confirm an event. Thus we begin modeling an anticipatory event at the sentence resolution.

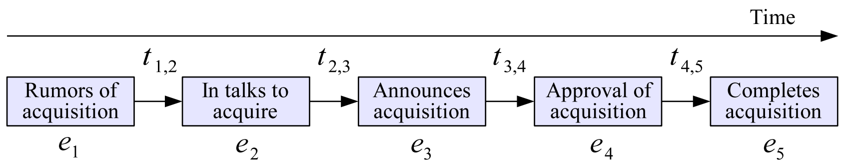


Fig. 2 Event Transition Graph for ''acquisition'' topic type

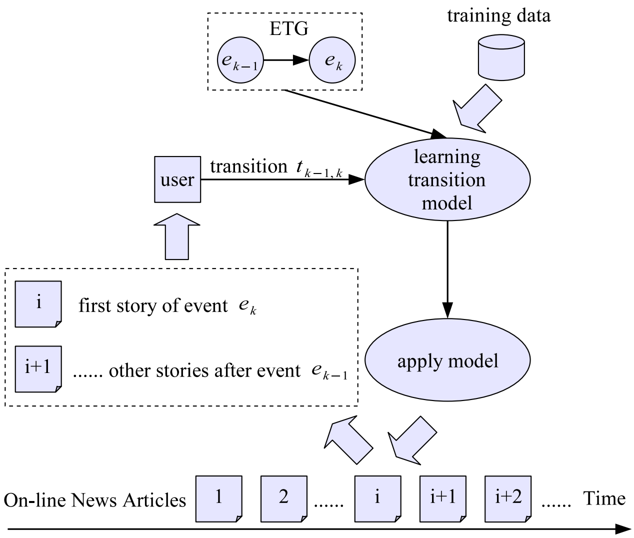


Fig. 3 Online AED system framework

In general, an 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’’.

1. Hit sentence: ‘‘The Knicks outscored Philadelphia 32-22 in the fourth quarter to secure the win.’’
2. User preference: ‘‘win basketball match.’’

The user preference defines an event transition ‘‘win basketball match’’ in the event transition graph for the ‘‘basketball match’’ domain.

4.2 Single-level and two-level SVM sentence classifiers

For the sentence classification model, we proposed a simple single-level SVM (Svm-light, http://www.svmlight.joachims.org/) sentence classifier and a more sophisticated two-level hierarchical SVM sentence classifier. The single-level SVM sentence classifier simply classifies all sentences as either positive (i.e., on-topic and event confirming) or negative (i.e., on-topic but non-event confirming, and off-topic). The two-level SVM classifier attempts to distinguish sentences describing current winnings from those about historical winnings as these sentences could otherwise confuse the single-level sentence classifier, which is also responsible for distinguishing on and off topic sentences about winnings. The sentences about current and historical events are known as *positive* and *historical sentences*, respectively. For example, ‘‘the rejuvenated Celtics have won three straight since then and six straight at home overall’’ is a typical historical sentence, which is considered as ‘‘on topic’’ by a single-level classifier but hard to be identified as non-event confirming because the single level classifier is not trained to distinguish event confirming sentences from non-event confirming sentences. The two-level classifier is able to handle most confusing cases like this.

Figure 4 shows the structure of the two-level SVM classifier. The first level classifier aims to detect all on-topic sentences, and these include both positive and historical sentences. The second level classifier performs a refinement on the on-topic sentences to further classify them into positive or historical.

4.3 Named entities analysis

Named Entities were originally created for extracting information from unstructured text (Grishman and Sundheim 1996), which primarily includes extracting names of people, places, organizations, etc. Named entities are extremely useful for text understanding due to the unstructured properties of text. With named entities, it is possible for us to answer questions such as "what", "who", "where", and "where" of a specific event. Kumaran and Allan (2004) 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 context semantics. In our approach, we extracted the game scores and basketball team names as two types of named entities and added them as additional features. Some examples of these named entities are identified within " < > ″ and shown in Fig. 5.

We observed that most positive sentences contain at least one team name and one score. However, some historical sentences also contain a team name and score. For example, sentence 3 in Fig. 5 contains two team names and one score, but it is a negative sentence reporting mid-game scores. Therefore, isolated named entities may be used to enhance the feature vector of sentences (as in our approach), but cannot be used alone to detect an AE.

Fig. 4 Two-level SVM sentence classifier

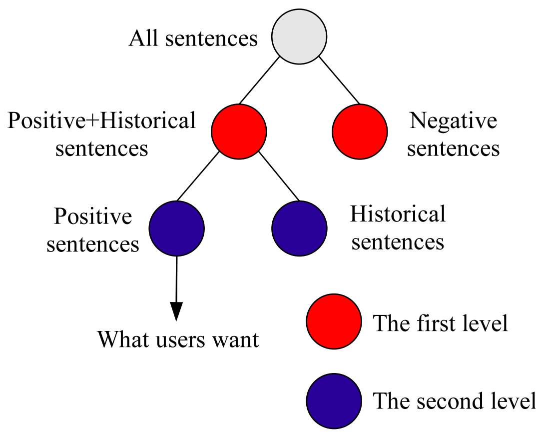


Fig. 5 Some typical named entities extracted in our AED system

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>...

4.4 Classification methods

We investigate various sentence retrieval strategies for AED, with a substantial focus on improving retrieval quality. In practice, the term weighting scheme used to represent a sentence vector has an enormous 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, TFISF, 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 introduction of ISF, called *inverse sentence frequency*. We also introduced the IDF, called *inverse document frequency*, at the sentence level to assume that terms appearing in a small number of documents are useful. The various term weighting schemes are summarized as follows:

$$\text{Standard TF} : f_{ij}, \tag{1}$$

$$\text{TFIDF} : f_{ij} \times \log\left(\frac{N}{n_i}\right), \tag{2}$$

$$\text{TFISF} : f_{ij} \times \log\left(\frac{S}{s_i}\right), \tag{3}$$

where f_{ij} is the frequency of term i in sentence j, N is the total number of documents in the collection, S is the total number of sentences in the collection, n_i is the number of documents containing term i, and s_i is the number of sentences containing term i. Our proposed weighting scheme, TF with named entities is simply standard TF appended with a team named entity feature and a score named entity feature denoting the frequencies of the two types of named entities, respectively.

4.5 Evaluation methodology

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

$$\text{Precision} = \frac{\#\ \text{correct positive predictions}}{\#\ \text{positive predictions}}, \tag{4}$$

$$\text{Recall} = \frac{\#\ \text{correct positive predictions}}{\#\ \text{positive examples}}. \tag{5}$$

To avoid "overfitting" the training data, we apply 10-fold cross validation and compute the average precision and recall for each method.

The harmonic mean of the average precision and recall (F-Measure), computed as shown below, provides the overall AED accuracy,

$$\text{F-Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{6}$$

5 Document classification model for AED

5.1 Motivation

Sentence retrieval is a very difficult problem (Allan et al. 2003) by itself. This is because a single sentence contains neither enough information (curse-of-dimensionality) nor context to form a meaningful model. Thus, we proposed another document based classification model for AED (He et al. 2006b) since a document contains significantly more information. In addition, annotating a document is much more easier than annotating all sentences in a document, thus AED at the document level becomes much more practical. Lastly, we need a specific mathematical evaluation criteria for AED, which is absent in the sentence classification model.

5.2 Named entities analysis

We observed that in the context of document, it is more necessary to alleviate the divergence brought by named entities instead of enhancing the context information by adding named entities as extra features. For example, consider the following two statements from two different news articles reporting the "announce acquisition" event:

"***China***'s biggest computer maker, ***Lenovo Group***, said on ***Wednesday*** it has acquired a majority stake in ***IBM Corp***'s personal computer business in a deal worth a total value of ***US$1.75 billion*** (***S$2.86 billion***), one of the biggest ***Chinese*** overseas acquisitions ever."

"***SBC Communications*** on ***Monday*** announced plans to acquire ***AT&T*** in a ***$16 billion*** deal, a move designed to bolster ***SBC***'s sales to enterprise customers nationwide and give it new national and global networks."

The words in boldface are the named entities. To detect both statements as related to the event transition, we need to find some common features shared by them. Unfortunately, the named entities from the two statements, if used as direct features, will only decrease the similarity between the two statements, making it more difficult to train an accurate AED classification model.

Therefore we replace the named entities by their types to create more interesting generic features, i.e., the named entity *types*, shared by the two statements. For example, the monetary terms "US$1.75 billion" and "$16 billion" will be replaced by the *Money* type, giving us new relevant features to determine that the statements are actually about acquisition.

After NE replacement, the two statements become:

GPE's biggest computer maker, ***ORGANIZATION***, said on ***DATE*** it has acquired a majority stake in ***ORGANIZATION***'s personal computer business in a deal worth a total value of ***MONEY*** (***ORGANIZATION MONEY***), one of the biggest ***NATIONALITY*** overseas acquisitions ever.

ORGANIZATION on ***DATE*** announced plans to acquire ***ORGANIZATION*** in a ***MONEY*** deal, a move designed to bolster ***ORGANIZATION***'s sales to enterprise customers nationwide and give it new national and global networks.

Clearly, after NE replacement, these two examples become more similar, which invariably helps the classifier better model the event transition.

5.3 Classification methods

We tried three different feature representation methods and one classifier combining strategy to train the AED classifier, as follows:

CONTENT	Entire news content as features.
TITLE	Title as features.
1SENT	First sentence as features.
VOTING	Majority voting on above three classifier outputs.

The TITLE and 1SENT representations were inspired by the observation that human experts can usually decide if a news is a hit simply based on its first sentence and/or title. Moreover, the TITLE and 1SENT representation of a news article may not always carry useful features, and the AED decision will have to fall back to the CONTENT representation. For example, the first sentence "*Signature Control Systems is off to a busy start in early 2006*" does not contain features really relevant to the "acquisition" event transition. VOTING was thus used as a simple and effective way to improve the overall accuracy.

5.4 Evaluation methodology

Suppose we are given a set of N news articles $X = \{x_1, ..., x_N\}$ about a topic, and an event transition graph $E = \{e_1, ..., e_n\}$ comprising n events. Each news x_i is assigned a publication date/time represented by $t(x_i)$ and an event type in E represented by $e(x_i)$, the latter of which is also known as the true event of x_i.

We assume that all news articles in X are sorted in chronological order, i.e., $t(x_i) \leq t(x_j)\ \forall\ i < j$, and all events in E are sorted in time, i.e., $t(e_i) \leq t(e_j)\ \forall\ i < j$.

By applying our trained AED classifier on a news article x_i, we obtained its assigned event denoted by $s'(x_i)$. Given a transition $t_{k-1,k}$ (i.e., user preference), the objective of AED is therefore to find the news article x_m that satisfies:

$$x_m = \arg\min \{t(x_i) \mid \forall x_i \text{ where } s'(x_i) = e_k\}.$$

To make the time comparison easier between the detected first story x_m and the event e_k, we also define the *true time* of e_k, $t(e_k)$, as follows:

$$t(e_k) = \min \{t(x_i) \mid \forall x_i \text{ where } e(x_i) = e_k\}.$$

Once the first story x_m of the anticipatory event e_k is determined by the AED classifier, all subsequent news articles, $x_j, j = (m + 1), ..., N$ will be assigned to event(s) e_k post transition $t_{k-1,\ k}$. Occasionally, the first story identified by AED may be prematured, delayed, or undefined (never found). Accordingly, we define four evaluation criteria as follows:

Accurate Alarm	$t(x_m) = t(e_k)$. First story of e_k found successfully.
Delayed Alarm	$t(x_m) > t(e_k)$. First story found was too late.
False Alarm	$t(x_m) < t(e_k)$. First story found was prematured.
Miss	$t(x_m)$ = undefined. No x_i in X has $s'(x_i) = e_k$. AED fails to even identify the event!

Figure 6 graphically depicts each of the four evaluation criteria for AE *Transition Detection*. Specifically, we simply tally the total number of *false alarms*, *delayed alarms*, *accurate alarms*, and *misses* to evaluate the AED performance on a given set of events. For news alerts, an *accurate alarm* is the most desirable, followed by a *delayed alarm*. Otherwise, a *miss* is generally preferred over a *false alarm*.

6 Anticipatory event dataset

We picked "basketball matches" as the topic of interest for the sentence classification model and "mergers and acquisitions" as the domain for the document classification model, respectively. Since AED is a new area introduced by us, with no standard evaluation benchmark dataset, we created our own datasets: *Basket100*, *Google Acquisition*, and *Acquisition7*.

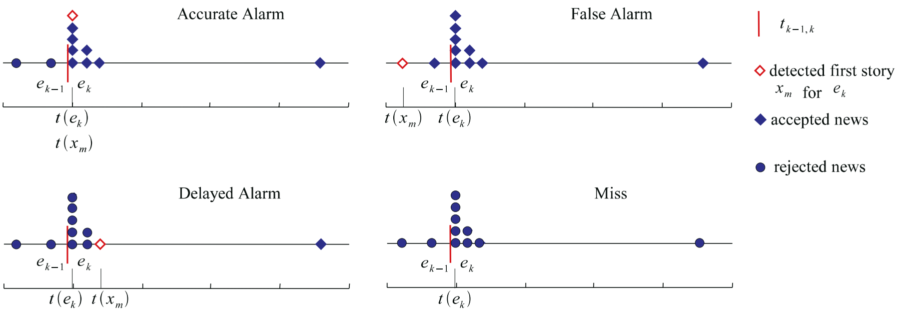


Fig. 6 An evaluation example for *Transition Detection* in AED

6.1 Basket100 dataset

The *Basket100* collection comprises 100 documents returned by Google using the user preference "win basketball match". In *Basket100*, 93 out of 100 documents are relevant, i.e., describes basketball games, and the remaining seven are irrelevant. The collection contains 2,340 sentences, comprising 4,499 unique terms (words). The 2340 sentences were manually annotated into three 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 1 shows the summary statistics for *Basket100*.

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 2. The sparsity metric is defined as:

$$\text{Sparsity} = \frac{T - A}{T}, \tag{7}$$

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

Table 1 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

Table 2 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%

From Tables 1 and 2, 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.

6.2 Google acquisition dataset

Short of automatically generating the training dataset for learning the event transition detection, we manually collected the *Google Acquisition* dataset, which contains 346 as-it-happens news articles returned by Google News Alerts during the 2-month period from December 19, 2005 to February 19, 2006 using the user preference ‘‘announce acquisition’’.

In *Google Acquisition*, 178 documents were manually labelled as positive and 168 as negative with respect to the ‘‘announce acquisition’’ transition, which means that Google News Alerts returned 168 (48.6%) outright false alarms for the subscribed keywords ‘‘announce acquisition’’. This level of retrieval quality is quite typical for simplistic keyword-based news alert system.

6.3 Acquisition7 dataset

The *Acquisition7* dataset contains news articles covering seven recent acquisition topics. It was created as the test data for the document classification model. Each acquisition news topic in *Acquisition7* is comprised of 20 news articles returned by Google News, approximately half of each (10) were reported before and after the ‘‘announce acquisition’’ transition.

Table 3 Make up of the *Acquisition7* dataset

Acquisition topics	$t(e)$
Adobe acquires Macromedia	April 18, 2005
CNPC acquires PetroKazakhstan	October 26, 2005
eBay acquires Skype	September 12, 2005
Lenovo acquires IBM PC Division	December 08, 2004
Oracle acquires PeopleSoft	December 13, 2004
Oracle acquires Siebel	September 12, 2005
SBC acquires AT&T	January 31, 2005

The seven acquisition news topics are listed in Table 3, where $t(e)$ refers to the true occurrence date for the “announce acquisition” event.

7 Experiment results

7.1 Experiment setup

Lucene 1.4.3 (Apache lucene 1.4.3, http://lucene.apache.org) was used to tokenize the news text content with stop word removal to create the corresponding 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 software package SVM-light (Svm-light, http://svmlight.joachims.org/) was used to build the various classification approaches. SVM cost factors (Morik et al. 1999) were used to offset the slight imbalance in numbers between the positive and negative documents.

7.2 Experiments on sentence classification model

The sentence model was the first solution proposed by us to solve the AED problem. The gathering of training documents can be viewed as a form of user preference, albeit an extremely laborious process. Two classifiers using various term weighting schemes of the sentence model were applied to the *Basket100* dataset, with the goal of detecting a winning basketball event.

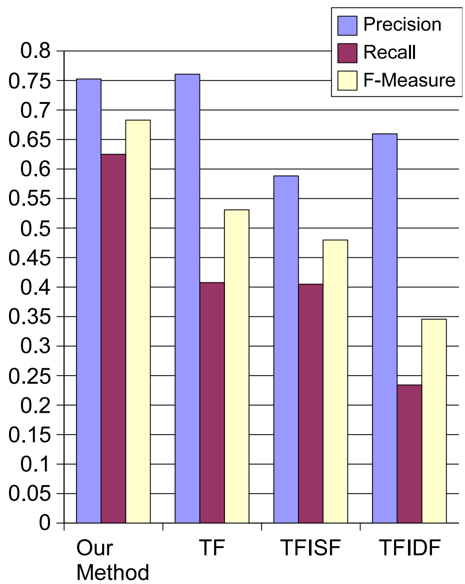


Fig. 7 Cross validated (10-fold) results of single-level SVM classifier

7.2.1 Single-level SVM sentence classifier

Figure 7 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 nearest competitor by 15%. Moreover, the recall of 0.63 is still 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.

7.2.2 Two-level SVM sentence classifier

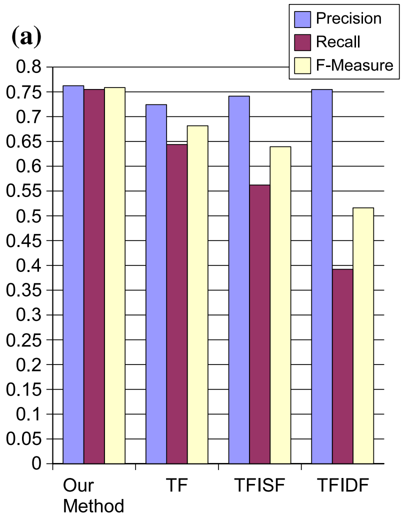
Figure 8 shows the results of the two-level classifier using different term weighting schemes. Since the first level classifier is only responsible for distinguishing on-topic sentences from off-topic ones, its performance was measured based on all on-topic sentences which included historical sentences.

Figure 8a, b shows 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 Fig. 8c, with our method achieving the overall best result of 0.69 precision and 0.72 recall.

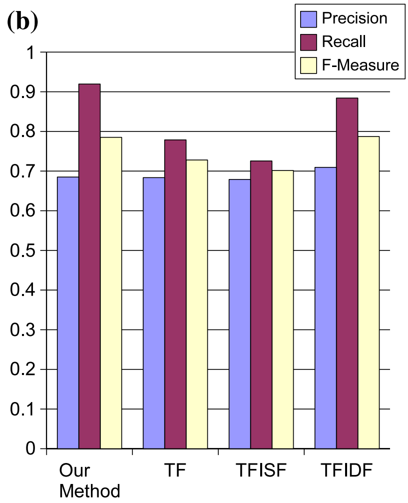
7.3 Discussions

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 the case in our experiments. Clearly, the first level classifier is crucial to the success of the second level classifier, and thus the overall accuracy.

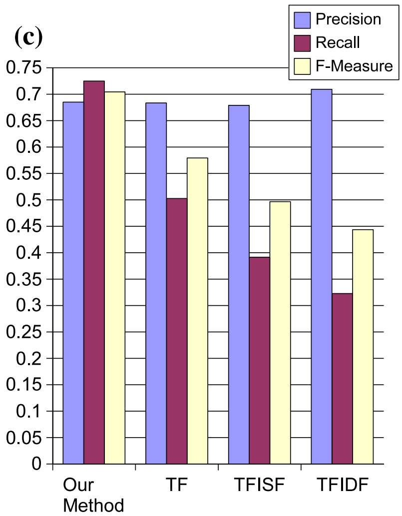
Figures 7 and 8c also show the F-Measure results for the single and two level classifiers using various term weighting schemes. Apparently, a two-level sentence classifier based on our named entity enhanced TF weighting yields the best overall performance in terms of F-Measure.



Test accuracy at first level.



Test accuracy at second level.



Overall test accuracy.

Fig. 8 Cross validated (10-fold) results of the two-level SVM classifier

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., waken up by more irrelevant alerts), he stands to catch 10% more (recall) of the actual alerts by using the two-level classifier. Thus, each approach has its pros and cons, and the ultimate choice is best left to the news alert subscriber.

7.4 Experiments on document classification model

After the limited success with Sentence AED Model, we refined the AED framework by analyzing whole documents instead of individual sentences.

7.4.1 Validating google acquisition dataset

In order to validate the generic "announce acquisition" trained model, we conducted 2-fold cross-validated experiments using the four text classification approaches of Sect. 3 on the *Google Acquisition* dataset. The dataset is first split along the timeline into two equal parts: (1) news articles dating from December 19, 2005 to Janary 19, 2006, and (2) news articles dating from January 20, 2006 to February 19, 2006. One part was used for training with the other part used for testing and vice-versa.

The significance of this experiment shown in Table 4 is that it increased the precision of Google's returned news alerts from 51.4 to 85.7%, a more than 33% improvement! All in all, the high precision and recall figures confirmed that the *Google Acquisition* dataset is indeed suitable for modelling "announce acquisition" event.

Table 4 Average test results on *Google Acquisition*

Average	CONTENT	TITLE	1SEN	VOTING
False alarms	22.5	15.5	17	**13.5**
Misses	**9**	24.5	15	10
Precision	0.7847	0.8110	0.8172	**0.8571**
Recall	**0.9011**	0.7308	0.8352	0.8901
F1	0.8389	0.7688	0.8261	**0.8733**

Best results shown bold

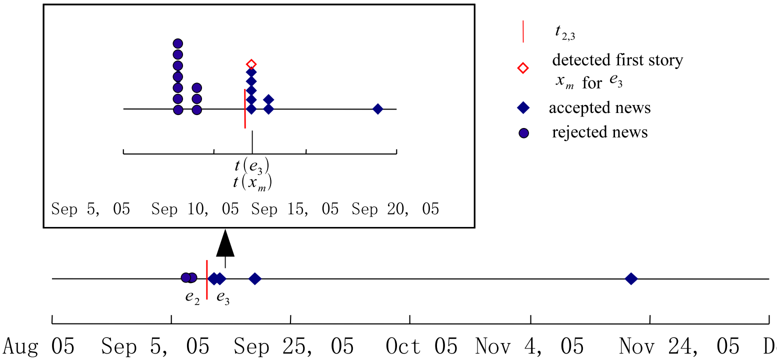


Fig. 9 AED on "eBay acquires Skype" found an accurate alarm, $t(x_m) = t(e_3)$

Table 5 AED results on *Acquisition7* using the VOTING method

Alarms	Accurate	Delayed	False	Miss
Adobe acquires Macromedia	√			
CNPC acquires PetroKazakhstan	√			
eBay acquires Skype	√			
Lenova acquires IBM PC Division			√	
Oracle acquires PeopleSoft			√	
Oracle acquires Siebel	√			
SBC acquires AT&T		√		

7.4.2 AED document classification model

In this section, we test the generic AED classifier trained by *Google Acquisition* on the *Acquisition7* dataset. One AED outcome is shown in Fig. 9. Note that once the ''first'' story of ''announce acquisition'' event has been identified by AED, all subsequent news articles are labelled ''positive''.

Table 5 gives a summary of the overall performances, which shows that AED based on the VOTING method generated four accurate alarms, one delayed alarm, two false alarms, and zero misses. This means that the model trained by *Google Acquisition* was able to cover the main characteristics of all seven acquisition topics.

8 Conclusion and future work

8.1 Conclusion

We proposed a new and practical application called anticipatory event detection (AED), which is a more refined and personalized form of event tracking and detection. We then investigated two classification methods to tackle the AED problem, a sentence AED model and a document AED model.

For the sentence classification AED model, we tested two approaches, a flat and two-level SVM models. In the process, we discovered that sentence classification performance is affected greatly 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 semantic structure into the classifier model, i.e., by using two layers, 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 (Allan et al. 2003). 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.

Several new contributions in the AED document classification Model are made as we (1) defined and formulated a mathematical model for the AED problem, (2) proposed a new way of applying named entities for AED, and (3) proposed a principled way to assemble generic training data for learning one AET, using the user's AE preference. The encouraging results obtained using our proposed document AED model hint at its practicality, and thus pave the way for additional future work.

8.2 Future work

The main limitation of AED lies in its reliance on a pre-trained 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., terrorist bombings, capture of terrorists, war, etc. This is acceptable if we can train a large list of AED models satisfying 80% of the ISI community.

In fact, we are currently collecting and generating several ISI topics to be analyzed by our AED framework. One of the ISI topics include the capture of important terrorists. For this ISI topic, the AED system could be trained using positive and negative news from the capture of Panama's Noriega and Iraq's Hussein,[1]. The test/evaluation set could include the capture of Abu Musab al-Zarqawi (killed), Abu Zubaydah (captured), Osama bin Laden (at large), etc. Currently, we are still refining the dataset and adjusting the AED framework; results will likely be reported in the near future.

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 represents a major development milestone in natural language understanding.

For the foreseeable future, we envisage a real-time feedback AED system that prompts an ISI analyst to refine his/her anticipatory event definition using similar historical events. For example, to define an anticipatory event such as "Osama bin Laden captured", the user can specify a similar transpired event like "Saddam Hussein captured", and the system shall return a list of historical documents for the user to label as positive/negative training documents. At its currently accuracy of around 70%, the AED system can already be put to practical use as an automated way to track important events. If and when the AED system achieves close to 90% accuracy with future improvements, it could very well become a truly effective alert system that ISI analysts can tailor to automatically gather open source intelligence.

References

Allan J (2002) Topic detection and tracking: event-based information organization. Kluwer, Dordrecht

[1] Note that these two events represent villified former head of states, not terrorists per se and thus may not provide relevant training for our AED system to detect terrorists capture.

Allan J, Jin H, Rajman M, Wayne C, Gildea D, Lavrenko V, Hoberman R, Caputo D (1999) Topic-based novelty detection: final report. In: Proceedings of the DARPA broadcast news transcription and understanding workshop
Allan J, Lavrenko V, Jin H (2000) First story detection in tdt is hard. In: Proceedings of the 9th ACM conference on information and knowledge management (CIKM), pp 374–381
Allan J, Wade C, Bolivar A (2003) 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, pp 314–321
Brants T, Chen F, Farahat A (2003) A system for new event detection. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval, pp 330–337
Chen Y, Crawfordand MM, Ghosh J (2004) Integrating support vector machines in a hierarchical output space decomposition framework. In: IEEE international geoscience and remote sensing symposium, vol II, pp 949–952
Chua K, Ong W-S, He Q, Chang K, Kek A (2005) Intelligent portal for event-triggered sms alerts. In: Proceedings of the 2005 IEE mobility conference
Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
Grishman R, Sundheim B (1996) Message understanding conference—6: a brief history. In: Proceedings of the 16th international conference on computational linguistics, pp 466–471
He Q, Chang K, Lim EP (2006a) Anticipatory event detection via sentence classification. In: Proceedings of the 2006 IEEE international conference on systems, man, and cybernetics, pp #01512
He Q, Chang K, Lim EP (2006b) A model for anticipatory event detection. In: Proceedings of the 25th international conference on conceptual modeling (ER), vol 4215, Springer LNCS, pp168–181
Jin H, Schwartz R, Sista S, Walls F (1999) Topic tracking for radio, tv broadcast, and newswire. In: Proceedings of the DARPA broadcast news workshop, pp 199–204
Kumar S, Ghosh J (1999) GAMLS: a generalized framework for associative modular learning systems. In: Proceedings of 1999 SPIE conference on applications and science of computational intelligence II, SPIE Proc. vol 3722, pp 24–35
Kumaran G, Allan J (2004) 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, pp 297–304
Lawrie D, Bruce Croft W (1999) Discovering and comparing topic hierarchies. In: Proceedings of the RIAO 2000 conference, pp 314–330
Morik K, Brockhausen P, Joachimss T (1999) 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), pp 268–277
Nallapati R, Feng A, Peng F, Allan J (2004) Event threading within news topics. In: Proceedings of the 13th international conference on information knowledge management (CIKM), pp 446–453
Quiggin T (2006) 'Dark caves' of secrecy can hinder intelligence. The Straits Times, Singapore, pp S12, Sep 30, 2006
Stokes N, Carthy J (2001) 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, pp 424–425
Strader TJ, Tarasewich P, Nickerson RC (2004) The state of wireless information systems and mobile commerce research. Inf Syst E-Bus Manage 2(4):287–292
Sun A, Lim EP (2001) Hierarchical text classification and evaluation. In: Proceedings of the 2001 IEEE international conference on data mining (ICDM), pp 521–528