An Aging Theory for Event Life-Cycle Modeling

Chien Chin Chen, Yao-Tsung Chen, and Meng Chang Chen

***Abstract*—An event can be described by a sequence of chronological documents from several information sources that together describe a story or happening. The goal of event detection and tracking is to automatically identify events and their associated documents during their life cycles. Conventional document clustering and classification techniques cannot effectively detect and track sequential events, as they ignore the temporal relationships among documents related to an event. The life cycle of an event is analogous to living beings. With abundant nourishment (i.e., related documents for the event), the life cycle is prolonged; conversely, an event or living fades away when nourishment is exhausted. Improper tracking algorithms often unnecessarily prolong or shorten the life cycle of detected events. In this paper, we propose an aging theory to model the life cycle of sequential events, which incorporates a traditional single-pass clustering algorithm to detect and track events. Our experiment results show that the proposed method achieves a better overall performance for both long-running and short-term events than previous approaches. Moreover, we find that the aging parameters of the aging schemes are profile dependent and that using proper profile-specific aging parameters improves the detection and tracking performance further.**

***Index Terms*—Clustering, knowledge life cycle, web mining.**

I. Introduction

THE WEB has become an important information source to serve all walks of life for various purposes. Via a variety of handy web composers, Internet users can easily act as information sources by publishing and sharing valuable knowledge. However, as the number of the web documents grows, obtaining desired information becomes increasingly time consuming and often requires specific knowledge to make the best use of search engines and the returned results. When an important event occurs, many information sources publish documents with different viewpoints of the event at different times. The hundreds of thousands of documents thus produced make it difficult for people to assimilate the full story of the event using existing Internet search tools. In many cases, alerts are required when some specific event occurs to allow users to make prompt responses, while summaries give readers a quick complete view of the events. Traditional search engine techniques cannot provide the needed functions; however, automatic event detection and tracking mechanisms can alleviate such difficulties.

The first study of event detection that attracted a great deal of attention was the Defense Advanced Research Projects Agency (DARPA) Topic Detection and Tracking (TDT) project [1], which considered event detection as a process of automatically identifying sequential events from online news streams. Here, an event is defined as something that occurs at a specific place and time and is associated with specific actions. Furthermore, it is reported in a sequence of chronological documents. In a sense, an event is similar to a category in traditional text classification. However, the characteristics of life cycles and context shifting distinguish events from categories. Unlike categories in text classification, which represent permanent topics in the real world, an event can be said to have a life cycle with the stages of birth, growth, decay, and death. In other words, an event fades away when few documents report or discuss it. Clearly, events have different life spans, depending on the degree of importance. Important events can last several weeks, while flash events may vanish within a few days. Another aspect of the life cycle of events is in terms of the number and distribution of documents that cover it [13]. Some events are supported by a large number of related documents, while others have only a few. Some events receive the most coverage at the start of the event, and some have relatively uniform coverage during their life cycles. These characteristics, coupled with the uncertainty of life cycles, make automatic event detection a challenging task.

Another difficulty of event detection is context shifting. During the life span of a sequential event, the themes of supporting documents may change frequently. For instance, for a sports tournament, pregame reports focus on the history of the event, records, and analyses of players; while postgame reports focus on recaps, critiques, and even fan reactions. Overlooking the issue of context shifting breaks up the storyline of an event so that users cannot get the whole picture.

Some previous works [3]–[30] addressed these crucial problems and showed that temporal information of sequential events can be used to discriminate events efficiently. In this paper, we present an aging theory to model the life cycles of sequential events. This theory deals with the context-shifting problem as well as the number and distribution of supporting documents, which make event life-cycle tracking more adaptive. Our experiment results show that the proposed approach can rectify the deficiencies of other methods.

The remainder of this paper is organized as follows. In Section II, we give a review of related works. Our aging theory is proposed in Section III. Section IV describes the algorithms used to adapt the aging theory to an event detection system. The performance of our method is evaluated in Section V. Finally, in Section VI, we present our conclusions and indicate the direction of future work.

Manuscript received January 1, 2005; revised August 14, 2005 and September 13, 2005. This work was supported in part by the National Science Council under Grant 93-2524-S-001-001 and Grant 94-2524-S-001-001. This paper was recommended by Associate Editor H. Chen.

C. C. Chen and M. C. Chen are with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, R.O.C. (e-mail: paton@iis.sinica.edu.tw; mcc@iis.sinica.edu.tw).

Y.-T. Chen was with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, R.O.C. He is now with the Department of Computer Science and Information Engineering, National Penghu University, Penghu, Taiwan, R.O.C. (e-mail: ytchen@npu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2006.886370

II. Related Works

Techniques, such as autoclassification and personalization, have been developed to help people access Internet documents. Classification systems, like ACIRD [18], Maria and Silva's [20], WPCM [26], and that of Wu *et al.* [29], build text classifiers from training data to categorize documents. Thus, users only need to read documents in the relevant categories to find the desired information. While the results in [18] show that categorized text improves retrieval performance, two problems make categorization impracticable for online event detection and tracking. First, the predefined and fixed set of categories cannot cover the constantly changing domain of events. Second, the training data of sudden events may be too small to acquire accurate classifiers. Moreover, readers usually follow events by their development threads, not by categories. For this reason, personalization is a useful mechanism to assist people in accessing Internet documents. Personalization systems [6], [8], [16] analyze readers' access patterns to create user profiles [14], which are then used to filter out irrelevant documents; hence, only the desired documents are recommended to users. Even though personalization can provide users with documents of interests, unexpected and not-to-be missed events, such as accidents, awards, and rainstorms, are sometimes omitted from suggestion lists because they rarely occur in an access pattern [8]. In addition, developments in short-term events may alter user interests, which often make personalization fallible [8]. The above discussion shows that existing systems would be enhanced by an efficient event detection algorithm.

TDT [1] is a DARPA-sponsored activity that processes information from streams of broadcast news. The project consists of five major tasks: story segmentation, topic tracking, topic detection, first story detection (FSD), and story link detection (SLD). The segmentation task locates the boundaries between adjacent stories in a news stream. Since, to TDT, the raw data represent streams of broadcast recordings and transcripts, segmenting the streams into constituent stories is necessary before performing the other tasks. Topic tracking [11], [17], [31] identifies documents about the same event. In some ways, topic tracking is similar to traditional information filtering, in that, given a set of target documents (or query terms), dissimilar documents are filtered out in sequence. However, because of the temporal characteristics of sequential events, topic tracking is more difficult than information filtering. Furthermore, the small number of target documents (between 1 and 16 in the TDT1 contest [4]) makes the topic tracking more challenging than information filtering. FSD, also known as online detection, identifies the first documents of events in a news stream. That is, whenever a document arrives, the FSD system must make a YES/NO decision to indicate whether the document initiates a new event. During the decision-making process, only previously examined documents are consulted to make the judgment. For this reason, the FSD is considered the hardest job in TDT [5]. SLD decides whether two randomly selected stories are about the same event. The SLD is considered the core operation of TDT [7], because when associations among documents can be precisely determined, other tasks like the FSD can be processed efficiently.

Event detection is similar to data corpus partitioning, which is usually solved by the traditional hierarchical agglomerative clustering (HAC) algorithm [25], that both partition a collection of documents into clusters. Depending on the similarity metric, a cluster or detected event in HAC can be regarded as: 1) a set of related documents; or 2) a centroid, formed by aggregating the content of clustered documents. One major problem with HAC is its high computation cost, which is quadratic to the number of input documents when using an average-link similarity metric [30]. This makes it impracticable when the data corpus is large. Besides the computation cost, the HAC-based methods are infeasible for environments where documents are generated constantly, such as online web news.

A popular event detection approach for online environments is single-pass (or incremental) clustering [4], [23], [27], whereby a single-pass clustering algorithm processes documents in a corpus sequentially. Similar to the HAC-based approach, a cluster in the single-pass clustering algorithm can be represented as a set of related documents or as an aggregated centroid. A document is merged with a cluster if the content similarity between them is above a predefined threshold; otherwise, the document is treated as the first document of a new cluster. Given two documents, the more keywords that co-occur, the more likely they are about the same event. However, in the semantics of event, the documents about the same event have a temporal association. For instance, two documents about car accidents have a similar wording, but they are likely not about the same event, if their publication time is months apart. Thus, including documents into clusters just according to content similarity may merge context similar, but unrelated documents together so that many important events may be missed.

In order to obtain better event detection and tracking results, the temporal relationships between documents and clusters must be incorporated into the clustering algorithm. Allan *et al.* [3] propose a time-based threshold approach to incorporate temporal information. They argue that temporally approximate documents are likely to discuss the same event. Hence, by constantly rising the similarity threshold of detection in time increments, it is unlikely that remote documents will be merged with existing clusters. Therefore, similar but different events can be identified easily. Meanwhile, Yang *et al.* [30] model the temporal relation using a time window and a decaying function. The size of the time window specifies the number of prior documents (or events) to be examined when clustering. The decaying function weights the influence of a document in the window based on the gap between the current document and an examined document. Similar to the time-based threshold approach, remote documents in the time window have less impact on clustering than those nearby. Recently, the techniques of natural language processing have been applied for event detection. Since an event usually refers to personal names, location names, time, etc., Makkonen *et al.* [19] analyze the sentence structure and extract proper nouns to discriminate context-similar events.

However, although the above methods can improve event detection results, they are not adaptable to different types of sequential events. Generally, an event is called short term if it vanishes quickly, and long running if it lasts for a longer

TABLE I
RECENT EVENT DETECTION METHODS

Method	Clustering Method	Event representation	Using temporal information
Yang's HAC [30]	HAC	Depends on the clustering metric	No
Allan et al [3]	Single-pass	Centroid	Yes
Yang's time window [30]	Single-pass	A set of related documents	Yes
Chen et al [9]	Single-pass	Centroid	Yes
Makkeone et al [19]	Single-pass	Centroid	Yes

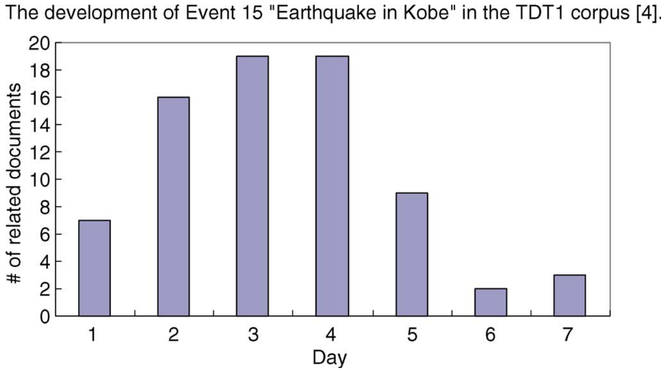

Fig. 1. Development of the event "Earthquake in Kobe."

time. Yang *et al.* [30] show that the HAC-based method does not work for long-running events. Since documents of a long-running event are usually scattered over a long period of time, there is a chance that the clustering mechanism cannot group related documents together. The time-based threshold method is also inappropriate for long-running events, because the increasing threshold prevents remote documents of such events from being tracked. While a larger window size can track the remote documents of long-running events, it also erroneously intermixes expired short-term events. To balance the tradeoff between long-running and short-term events, a self-adaptive event life-cycle management mechanism is needed. In the following section, we present an event detection and tracking method that employs the aging theory to model the life cycles of events. Table I summarizes the properties of some event detection methods.

III. AGING THEORY

In the online event detection and tracking environment, a sequential event is supported by a sequence of chronological documents and can be considered as a life form that develops through the aging stages of birth, growth, decay, and death. Fig. 1 illustrates such a development with a real-world example. The figure designates the number of news documents per day of the event "Earthquake in Kobe" from the TDT1 corpus [4]. As we can see from the figure, the event gradually becomes popular in the first four days. After it loses attractions, it finally fades away.

The goal of aging theory is to provide a formal model of the aging behavior of a sequential event. We first present the definitions used in the theory, and then describe three aging models of the life span of events.

A. Definitions

The aging theory maps the development of an event into an energy value, which indicates the event's status and is used to predict its possible life span. Like the endogenous fitness of an artificial life agent [21], the energy value expresses the event's popularity. A high energy value implies that the event is popular, while a low energy value implies that it is out of favor. Analogous to the energy of a life form, the energy value of an event fades with time if it does not receive nourishment, i.e., supporting documents. We use the degree of the content similarity between a document and an event to indicate the energy support of the document to that event. In this way, documents about the same or a similar topic to the event contribute higher energy values than unrelated ones, just as foods generate different levels of nutrition for life forms.

To track the development of events, we view time as a series of time slots. For an event V, let x_t denote the total support from its supporting documents in a time slot t, i.e., the summation of similarities between the event and the supporting documents in time slot t. Let $y_t = g(x_1, \ldots, x_t, \alpha, \beta)$ be the accumulative support at time t, which is the return value of the function g with variables, including the support from time 1 to t and two aging parameters, α and β, $0 \leq \alpha \leq 1, 0 \leq \beta \leq 1$. The two aging parameters α and β represent the core of the aging theory that describes the life cycle of events. Parameter α, called the support transfer factor, decides the influence of documents on the life of an event. Parameter β, called the support decay factor, governs the pace of aging.

We present three aging schemes, growth only, constant decay (CD), and recursive decay (RD), to calculate an event's accumulative support. For growth only, the accumulative support $y_t = \sum_{i=1,\ldots,t}(\alpha x_i)$ is a direct summation of previous support. For CD, $y_t = \sum_{i=1,\ldots,t}(\alpha x_i - \beta)$ loses support at a constant pace of β for every time slot. For RD, $y_t = \alpha(\beta y_{t-1} + (1 - \beta)x_t)$ is a weighted recursion of the event's previous support. Briefly, the growth-only aging scheme is a trivial case of aging functions that the energy of events never fade away, while the schemes of CD and RD decrease the energy values periodically. Before elaborating on these aging schemes, we introduce the energy function, which converts the accumulative support of an event into an energy value. The energy function $F()$ must meet the following three conditions:

$$0 \leq F(y) \leq 1 \tag{1}$$

$$F(y) \text{ is a strictly increasing function of } y \tag{2}$$

$$F(\infty) = 1 \text{ and } F(0) = 0. \tag{3}$$

The energy function transforms the unlimited range [0, ∞) of the accumulative support y_t into a limited energy value within [0,1). With a limited range, we can give a meaning to each segment of the range and interpret the status of an event by its energy value. For example, let an energy value between 0.8 and 1.0 be defined as "active." Thus, if an event has an energy

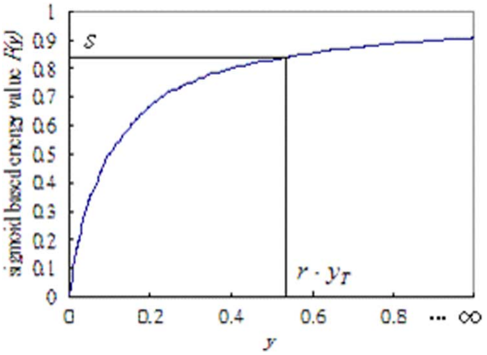

Fig. 2. Output curve of the sigmoid function.

value of 0.85, we can interpret it as active. Furthermore, with a bounded energy value, the conditions required for solving the aging parameters of the aging schemes can be much easier to be defined, which will be discussed in the subsequent sections. Note that the process of solving the aging parameters is performed on the accumulative support, rather than energy value, and is thus independent of the selected energy function. Therefore, the choice of energy function does not affect the detection results. In this paper, we adopt a sigmoid function as our energy function, which is defined as follows:

$$F(y) = 10y/(1+10y), \qquad y > 0$$
$$= 0, \qquad \text{otherwise.} \tag{4}$$

Aging schemes together with the aging parameters generate distinct aging profiles. The main issue for aging theory is to find an adequate combination of α, β, and the scheme of y_t such that the energy value of an event is one when all its supporting documents appear. However, by (3), the energy value of an event can never be one, as its support value can never be infinite. Therefore, we loosen the above statements and redefine the condition as

$$F(r \cdot y_T) = s \tag{5}$$

where $0 < r, s < 1, T$ is the number of time slots that the event V spans, and y_T is the accumulative support.

Intuitively, (5), as depicted in Fig. 2, can be interpreted as the energy function returns the energy value s when the r percentages of an event's supporting documents appear. For example, if $r = s = 0.85$, the acquired aging parameters cause the energy value to be 0.85 when 85% of an event's supporting documents appear. In the following sections, we discuss how to obtain the aging parameters α and β for the three aging schemes.

B. Growth-Only Aging Scheme

One trivial case of aging schemes is growth only, which means that the energy of a detected event never decays. In this case, β is zero and y_t is the accumulated support of all the event's supporting documents at time t. To find the value of the aging parameter α, we set $y_T = \sum_{i=1,\ldots,T}(\alpha x_i)$ in (5), and have

$$F\left(r \sum_{i=1,\ldots,T} (\alpha x_i)\right) = s. \tag{6}$$

According to the conditions (1) to (3), the energy function must be one to one and onto, and therefore has a corresponding inverse function F^{-1} [12]. We apply the inverse function F^{-1} to both sides of (6)

$$r \sum_{i=1,\ldots,T} (\alpha x_i) = F^{-1}(s). \tag{7}$$

Dividing both sides by $r\sum_{i=1,\ldots,T}(x_i)$, to solve α, we have

$$\alpha^* = F^{-1}(s)/r \sum_{i=1,\ldots,T} (x_i). \tag{8}$$

C. CD Aging Scheme

Since the energy value of the growth-only aging scheme never declines, detected events will not be eliminated during the process of event detection. Therefore, its detection result should be identical to that of the traditional single-pass clustering algorithm, which does not use any temporal information for more effective event detection. To emulate fading energy, the CD aging scheme subtracts a constant support β from the accumulative support for every time slot. In this way, events with little or no follow-up support documents will gradually fade. The accumulative support of the CD aging scheme is defined as

$$y_t = \sum_{i=1,\ldots,t} (\alpha x_i - \beta) = \alpha \sum_{i=1,\ldots,t} (x_i) - \beta t. \tag{9}$$

Equation (9) has two unsolved parameters α and β that requires two pairs of (r, s), called (r_1, s_1) and (r_2, s_2), from (5) to obtain the parameter values. However, applying the pairs into (5) directly results in two dependent equations, which gives infinite numbers of solutions for α and β. Therefore, we loosen (5) such that when the accumulated support reaches r percentages of the sum of all support, the energy value is s. Assuming t is the time slot in which the above condition holds, the above statements can be expressed as

$$F\left(\sum_{i=1,\ldots,t} (\alpha x_i - \beta)\right) = F\left(\alpha \sum_{i=1,\ldots,t} (x_i) - \beta t\right) = s. \tag{10}$$

As $\sum_{i=1,\ldots,t}(x_i) = r\sum_{i=1,\ldots,T}(x_i)$, (10) can be represented as follows:

$$F\left(\alpha r \sum_{i=1,\ldots,T} (x_i) - \beta t\right) = s. \tag{11}$$

Applying (r_1, s_1) and (r_2, s_2) and the corresponding time slots t_1 and t_2 to (11), we get the following new optimal conditions:

$$F\left(\alpha r_1 \sum_{i=1,\ldots,T} (x_i) - \beta t_1\right) = s_1 \tag{12}$$

and

$$F\left(\alpha r_2 \sum_{i=1,\ldots,T} (x_i) - \beta t_2\right) = s_2. \tag{13}$$

We then take the inverse function F^{-1} of both sides to get

$$\alpha r_1 \sum_{i=1,\ldots,T} (x_i) - \beta t_1 = F^{-1}(s_1) \tag{14}$$

and

$$\alpha r_2 \sum_{i=1,\ldots,T} (x_i) - \beta t_2 = F^{-1}(s_2). \tag{15}$$

Solve α and β from (14) and (15)

$$\alpha^* = \left[t_2 F^{-1}(s_1) - t_1 F^{-1}(s_2)\right] \Big/ \left[(r_1 t_2 - r_2 t_1) \sum_{i=1,\ldots,T} (x_i)\right] \tag{16}$$

and

$$\beta^* = \left\{r_1 \left[t_2 F^{-1}(s_1) - t_1 F^{-1}(s_2)\right] / (r_1 t_2 - r_2 t_1) - F^{-1}(s_1)\right\} / t_1. \tag{17}$$

D. RD Aging Scheme

The RD aging scheme employs an exponential smoothing approach, called the RD, to model an event's energy decay. This approach is frequently used to analyze time-series data, such as network traffic control [28], and to uncover the trend of data movement. In an RD scheme, the accumulative support of an event at time t is a weighted combination of its preceding support and the support contributed by the supporting documents in the current time slot. The formal definition of the y_t is

$$y_t = \alpha\left[\beta y_{t-1} + (1-\beta)x_t\right], \qquad \text{where } y_0 = 0. \tag{18}$$

Similar to the CD aging scheme, events with little or no follow-up support gradually die out. By substituting y_i with the function of x_i and y_{i-1} and so on, the above equation can be unfolded as a combination of $x_1, \ldots, x_t, \alpha$, and β without the recursive term

$$y_t = (1-\beta)\left[(\alpha\beta)^{t-1}x_1 + (\alpha\beta)^{t-2}x_2 + \cdots + x_t\right]. \tag{19}$$

As we can see from the unfolded (19), the high orders of α and β make the approach used in the last section infeasible. Again, we loosen the condition as below and transform the parameter acquisition problem into the constraint satisfaction problem (CSP) [15].

$$F\left(\mathrm{MIN}\{y_1, y_2, \ldots, y_T\}\right) \geq s_1 \tag{20}$$

and

$$F\left(\mathrm{MAX}\{y_1, y_2, \ldots, y_T\}\right) \geq s_2 \tag{21}$$

where

$$1 \geq s_2 \geq s_1 \geq 0.$$

The reason for choosing these two criteria is to define the fading and qualifying thresholds of an event. That is, s_1 is an energy threshold that removes fading events and s_2 indicates that a likely event is qualified. With the conditions in (20) and (21), the acquired aging parameters guarantee that an event has sufficient and necessary energy during its lifetime. After converting the parameter acquisition problem into CSP, we adopt the numerical method in Marc Torrens's Java Constraint Library [2] to approximate the aging parameters.

E. Training of α and β

In the above sections, the solutions of the aging parameters are given for each aging scheme. To put the aging theory into practice, we first collect a set of training events. After deciding the values of the objective conditions (r_1 and s_1 for the growth-only scheme, r_1, s_1, r_2, and s_2 for the CD scheme, and s_1 and s_2 for the RD scheme), the above equations are used to find adequate aging parameters for the training events. Then, we use the averages of the parameters obtained from the training events as the learned parameters which are applied in the following proposed event detection and tracking algorithms.

IV. Event Detection and Tracking

Aging theory provides a mathematical foundation for modeling the aging behavior of chronological documents. In this section, we incorporate the aging theory into a single-pass clustering algorithm to detect events efficiently and introduce the data structures used in the algorithm.

A. Data Scheme of Events

Both documents and events are represented as a vector in the conventional vector space model (VSM) [25]. A vector in VSM is a set of weighted terms in which weights indicate the significance of terms in the context of the document (or the event). For documents, we use TF-IDF [25] for the term weight, which is defined as

$$w_{t,d} = tf_{t,d} \cdot \log(N/df_t) \tag{22}$$

where $w_{t,d}$ is the weight of term t in document d; $tf_{t,d}$ is the term frequency (TF) of term t in document d; $\log(N/df_t)$ is the inverted document frequency (IDF) of term t; N is the number of documents in the system's corpus; df_t is the number of documents in the corpus where t occurs.

TF-IDF determines the weight of a term by combining its TF and the IDF. In general, terms with high TF should weigh more. However, only considering the TF over the counts of general terms results in poorly distinguished documents. IDF alleviates this problem by reducing the weights of general terms, and experiments indicate that using TF-IDF ensures excellent indexing performance [25].

Since an event comprises a sequence of supporting documents, its term weights can be derived from the documents, but the weights vary with time. That is to say, the weights of terms representing an event should be updated progressively to reflect the event development. We use the Rocchio method [24] to update the term weights of an event incrementally. The equation is defined as

$$w_{t,e} = (1-\gamma) \cdot w_{t,e} + \gamma \cdot w_{t,d} \tag{23}$$

where $w_{t,e}$ is the weight of term t of event e; $w_{t,d}$ is the weight of term t in the inserted document d; γ, in [0, 1], is a parameter that adjusts the contribution of document d to event e.

Every time an event finds a supporting document, it updates its term weights by combining the original term weights with the weights of the new document. In this way, the term vector of the event can keep up with the development of the storyline. As well as the term vector, each event has a real number variable, denoted as eng, which indicates its energy. The energy of an event increases when it becomes popular, but decreases if few follow-up documents occur in the same time slot. Therefore, events that attract little interest will gradually disappear. Both the content and the status of an event can be described by term vectors and energy values, respectively. Moreover, as documents and events are all modeled in VSM, the similarity between them can be easily derived from the cosine value between the corresponding vectors.

B. Energy-Based Event Detection Algorithm

The energy-based event detection algorithm is listed in Fig. 3. Symbol E is a set of candidate events detected by the algorithm. Initially, E is empty. For each document d that occurs in time slot t, the similarity of d to the most similar event e in E is examined against a predefined threshold, called $\mathrm{threshold}_{\mathrm{detect}}$. If the similarity is greater than the threshold, the document is considered a support document to event e. Otherwise, we treat the document as a newly detected event by calling the function CreateNewEvent().

```
Energy-based Event Detection Algorithm:
  $E$ = null;
  for $t$ = 1 to $\infty$
    set $e.x_t = 0$ for all $e$ in $E$;
    for each document $d$ occurred in time slot $t$
      $e = \mathrm{ARGMAX}_{e \text{ in } E}(sim(e,d))$;
      if $sim(e,d) \geqq threshold_{detect}$ then
        $e.x_t = e.x_t + sim(e,d)$;
        $e.VectorUpdate(d)$;
      else
        $e_{new} = CreateNewEvent(d)$;
        add $e_{new}$ into $E$;
      end if
    end for
    for each event $e$ in $E$
      $e.EnergyUpdate()$;
      if $e.eng \leqq threshold_{remove}$ then
        remove $e$ from $E$;
      end if
    end for
  end for
```

Fig. 3. Energy-based event detection algorithm.

To capture the storyline development as well as the life span of event e when a document is associated with the event, we call the function VectorUpdate(), which implements the Rocchio method in (23), to update the event's term vector. Meanwhile, we add the document to event similarity to $e.x_t$ to record the total support of event e in time slot t. After all the documents in time slot t have been processed, the accumulated total support x_t together with the learned aging parameters α and β are used to update the event's energy value. For the CD scheme, the function e.EnergyUpdate() is defined as

$$\mathrm{e.eng} = F\left(F^{-1}(\mathrm{e.eng}) + \alpha \cdot e.x_t - \beta\right) \tag{24}$$

where $\mathrm{e.eng}$ is the energy value of event e;$F()$ is the energy function; $F^{-1}()$ is the inverse energy function; $e.x_t$ is the total support of event e in time slot t; α and β are the aging parameters learned from (16) and (17).

For the RD scheme, the function e.EnergyUpdate() is defined as

$$\mathrm{e.eng} = F\left(\alpha \cdot \left(\beta \cdot F^{-1}(\mathrm{e.eng}) + (1-\beta) \cdot e.x_t\right)\right) \tag{25}$$

where $\mathrm{e.eng}$ is the energy value of event e; $F()$ is the energy function; $F^{-1}()$ is the inverse energy function; $e.x_t$ is the total support of event e in time slot t; α and β are the aging parameters that satisfy (20) and (21).

The updated energy value is then compared with a predefined threshold, called $\mathrm{threshold}_{\mathrm{remove}}$. If the value is lower than the threshold, this outdated event is removed from the candidate set E.

V. Empirical Evaluations

We built a prototype system and conducted intensive experiments to evaluate the proposed aging theory. As the aging parameters α and β determine the performance of our energy-based event detection algorithm, in this section, we first investigate the influence of α and β on the detection results via two-factor ANOVA testing. Then, after showing that the aging parameters are significant in event detection, we demonstrate that the parameters acquired by our proposed theory outperform those of other selection methods. Finally, we compare our approach with three event detection methods [3], [4], [30]. The results show that our method outperforms the other methods for both long-running and short-term events.

A. Prototype System

As depicted in Fig. 4, news documents from several news agencies are directed to our system. Before running the

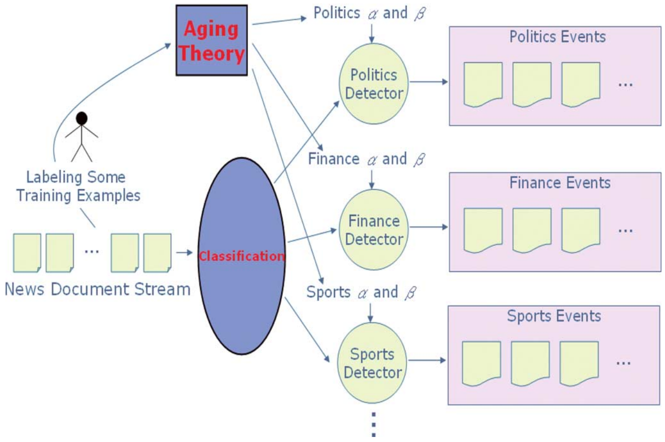

Fig. 4. System architecture based on aging theory.

TABLE II
STATISTICS OF THE DATA CORPUS

Start-end date	2002/10/1—2002/12/31		
Number of news documents		43,523	
Category	Politic	Sports	Entertainment
Number of labeled events	18	19	10
Event period ≦ 3	3	11	1
3 < Event period ≦ 7	5	7	3
7 < Event period	10	1	6

proposed event detection algorithm, each incoming document is first classified into its most appropriate category according to [18]. The reason for doing so is that events from different categories usually behave differently. Hence, by using profile-specific aging parameters, the event detection achieves a better performance. While in certain situations, categories can form a hierarchy or lattice to capture real-world semantics, in this paper, only a layer of categories is used.

B. Data Corpus and Evaluation

Table II details the corpus we collected for evaluation. It contains 43 523 local news documents from October 1, 2002 to December 31, 2002. Each document in the corpus is classified into one of the categories in [18]. To avoid classification errors in the data corpus, human experts examine the classification results and prepare the training and test datasets; 47 events are labeled by human experts from the politics, sports, and entertainment categories for evaluation. Events are identified as short term if they disappear within three days, and long running if they last over a week. Identifying the type of event allows us to discuss the advantages of each of the compared methods under different situations. It is worth noting that we composed this corpus to verify if the aging behavior of an event is profile dependent, which the popular TDT pilot study corpus [4] does not show. Nevertheless, we still compare our methods with the others based on the TDT corpus in the last experiment in this paper.

Table II shows that most political and entertainment events were long running, while only one sports event lasted longer than a week. This distribution supports our argument about profile-dependent aging behaviors that events satisfying certain criteria, such as classification, may have unique aging behavior.

TABLE III
EVENT CONTINGENCY TABLE

	In labeled event	Not in labeled event
In generated cluster	a	b
Not in generated cluster	c	d

$m = c / (a + c)$ if $a + c > 0$, otherwise undefined;
$f = b / (b + d)$ if $b + d > 0$, otherwise undefined;
$r = a / (a + c)$ if $a + c > 0$, otherwise undefined;
$p = a / (a + b)$ if $a + b > 0$, otherwise undefined;
$F1 = 2pr / (p + r)$
$cost = 0.98{*}f + 0.02{*}m$;

TABLE IV
NUMBER OF DETECTED EVENTS FOR THE CD SCHEME

Number of detected events (cluster)		α				
		0.1	0.3	0.5	0.7	0.9
β	0.1	6250	5927	5831	5807	5759
	0.3	7432	7085	7057	7026	6851
	0.5	8338	7561	7559	7561	7561
	0.7	8742	7691	7571	7559	7561
	0.9	9243	8960	8622	8406	8362

The performance of each method is evaluated as follows. First, each compared method partitions the data corpus into clusters (i.e., detected events), and the detected events which best matched human-labeled events are evaluated using the six TDT official metrics in [4]. The degree of similarity between a labeled event and a generated cluster is determined by the number of documents appearing in both the event and the cluster. Table III lists the cluster and event contingency together with the six official TDT metrics, namely, precision (p), recall (r), miss (m), false alarm (f), F1 measure ($F1$), and cost.

In the field of information retrieval, precision and recall are two important measures for evaluating the effectiveness of a clustering algorithm. High precision indicates that the system can generate coherent clusters, while high recall means that generated clusters precisely cover information about an event. Usually, these two metrics are inversely related; that is, when precision increases, the recall typically decreases and vice versa [10]. Therefore, only considering precision or recall yields biased results, whereas $F1$, which considers both precision and recall, is an objective measure. Similar to precision and recall, $F1$ ranges between zero and one. High $F1$ values indicate that the algorithm has good balanced precision and recall. By contrast, low $F1$ values mean that the algorithm either cannot cluster correctly or it cannot catch the whole story.

C. Significance of the Aging Parameters

To show the influence of the aging parameters on our energy-based event detection algorithm, we first employ a two-factor ANOVA test to analyze the significance of the parameters α and β. ANOVA is a statistical test that verifies whether a factor is significant to an experiment. We run the algorithm several times using various parameter combinations and record the number of detected events for the two-factor ANOVA test. Tables IV and V show the experiment results of the CD and RD

TABLE V
NUMBER OF DETECTED EVENTS FOR THE RD SCHEME

Number of detected events (cluster)		α				
		0.1	0.3	0.5	0.7	0.9
β	0.1	7113	6759	6732	6524	6492
	0.3	7047	6543	6320	6126	5914
	0.5	6837	6470	6092	5802	5510
	0.7	6809	6314	5905	5487	5038
	0.9	7024	6310	5805	5201	4519

TABLE VI
TWO-FACTOR ANOVA TABLE FOR THE CD SCHEME AGAINST THE NUMBER OF DETECTED EVENTS

Source of variance	Sum of squares	Degree of freedom	Mean square	F-value
β	21534153.04	4	5383538	199.7476
α	2018396.24	4	504599.1	18.72234
error	431227.36	16	26951.71	
total	23983776.64	24		

Level of significance: 0.05, Critical value: 3.006917

TABLE VII
TWO-FACTOR ANOVA TABLE FOR THE RD SCHEME AGAINST THE NUMBER OF DETECTED EVENTS

Source of variance	Sum of squares	Degree of freedom	Mean square	F-value
β	2901852	4	725463.1	9.127226
α	6503902	4	1625976	20.45679
error	1271735	16	79483.41	
total	10677489	24		

Level of significance: 0.05, Critical value: 3.006917

schemes, respectively. The corresponding ANOVA analyses in Tables VI and VII show that the F-values of α and β for the CD and RD schemes are all far above the critical value (3.006917). This indicates that the aging parameters indeed influence the result of our energy-based event detection algorithm.

Analyzing Table IV further, it is clear that the number of detected events grows when β increases or α decreases. A large β parameter in the CD scheme causes detected events to lose energy quickly. Note that a detected event may die before the actual end of the corresponding real-world event. Accordingly, a real-world event may be broken into several detected events, which leads to a large number of clusters. Similarly, a small α parameter in the CD scheme reduces the contribution of incoming documents to an event's energy value, which causes the detected events to be malnourished and die early. This also increases the number of clusters. With regard to the RD scheme, Table V shows that the number of detected events decreases when α or β increases. According to (18), the magnitude of α multiplied by β determines the pace of decay. A large α or β slows the speed of decay and thus results in a small number of clusters.

These experiments show that the aging parameters affect the result of event detection in both the CD and RD schemes. Hence, it is important to select the aging parameters carefully. The following experiments verify that the aging theory is capable of generating high-performance aging parameters.

TABLE VIII
EFFECTIVENESS OF THE AGING PARAMETERS OF THE CD METHOD

	p	*r*	*m*	*f(‰)*	*cost(%)*	*F1*
Category-specific parameters	**0.74**	**0.84**	**0.15**	**0.17**	**0.32**	**0.75**
Parameters from other categories	0.72	0.82	0.16	0.20	0.35	0.72
Random Selection	0.69	0.78	0.20	0.24	0.44	0.67

TABLE IX
EFFECTIVENESS OF THE AGING PARAMETERS OF THE RD METHOD

	p	*r*	*m*	*f(‰)*	*cost(%)*	*F1*
Category-specific parameters	**0.76**	**0.83**	**0.15**	**0.12**	**0.32**	**0.75**
Parameters from other categories	0.75	0.81	0.17	0.21	0.38	0.73
Random Selection	0.70	0.82	0.17	0.24	0.37	0.70

D. *Effectiveness of the Aging Theory*

In this section, we examine the effectiveness of the aging parameters obtained by different approaches. For aging-theory approaches, a cross-validation method [22] is employed to induce credible results. For each category, the corpus is divided into nine or ten sets, each of which contains at most two nonoverlapping labeled events. In each cross-validation run, one set is selected for testing, and the remaining sets are used for training to obtain the aging parameters, as described in Section III. Furthermore, to verify that each category has its own most appropriate aging parameters, we examine the detection performance for each category using the parameters learned from other categories. For the CD scheme (Table VIII), r_1 and s_1 are set at 0.15, and r_2 and s_2 are set at 0.95; and for the RD scheme (Table IX), s_1 is set at 0.001 and s_2 is set at 0.95. In addition to the aging theory, we randomly assign values to α and β to determine whether the detection performance of randomly assigned parameters is as good as the learned ones. The selection approach randomly assigns the aging parameters five times and averages the detection results for comparison. The following tables show the evaluation results.

The experiment results show that the CD and RD schemes work well with their own category-specific parameters. For the CD scheme, the average values of α and β for political events, sports events, and entertainment events, which are (0.14361, 0.11416), (0.52841, 0.32698), and (0.11544, 0.10065), respectively, faithfully describe the life spans of the events. A small α for the political and entertainment categories reveals that political and entertainment events are generally accompanied by a burst of documents. To avoid detecting a premature event, the CD scheme automatically adapts by reducing the energy contribution of supporting documents. In contrast, to catch light sports events, a larger α is necessary to boost the energy contribution and thus extends the life span. As shown in Table II, sports events usually vanish within three days; hence, a larger β value can appropriately speed up the aging process. Relatively, smaller β values for political and entertainment events cause appropriately longer life spans. For the RD scheme, the averages of α and β for political events, sports events, and entertainment events, which are (0.454898, 0.130158), (0.784185, 0.002975), and (0.47775,

TABLE X
EXPERIMENT RESULTS FOR POLITICAL EVENTS

	p	*r*	*m*	*f(‰)*	*cost(%)*	*F1*
B	0.66	0.70	0.29	0.32	0.62	0.64
CD	0.75	0.73	0.26	0.24	**0.55**	**0.70**
RD	0.78	0.72	0.27	0.23	0.56	0.69
T0.02	0.85	0.62	0.37	0.05	0.74	0.69
T0.05	**0.87**	0.58	0.41	0.05	0.83	0.67
T0.1	0.86	0.50	0.49	**0.04**	0.98	0.59
W2000	0.55	**0.79**	**0.20**	1.56	0.56	0.54
W2000d	0.79	0.65	0.34	0.31	0.71	0.66
W3000d	0.76	0.72	0.27	0.35	0.57	0.68

TABLE XI
EXPERIMENT RESULTS FOR SPORTS EVENTS

	p	*r*	*m*	*f(‰)*	*cost(%)*	*F1*
B	0.42	0.87	0.12	0.38	0.28	0.52
CD	0.63	0.93	0.06	0.22	0.15	0.71
RD	0.61	**0.94**	**0.06**	0.11	**0.13**	0.71
T0.02	0.68	0.81	0.18	0.09	0.37	0.71
T0.05	0.75	0.78	0.21	0.08	0.44	**0.74**
T0.1	**0.78**	0.73	0.26	**0.05**	0.54	0.72
W2000	0.55	0.93	0.07	0.45	0.18	0.62
W2000d	0.67	0.83	0.16	0.24	0.34	0.67
W3000d	0.64	0.88	0.11	0.31	0.26	0.67

TABLE XII
EXPERIMENT RESULTS FOR ENTERTAINMENT EVENTS

	p	*r*	*m*	*f(‰)*	*cost(%)*	*F1*
B	0.57	0.83	0.16	0.36	0.36	0.65
CD	0.85	0.86	0.13	0.07	0.27	0.84
RD	0.87	0.85	0.14	0.05	0.28	**0.85**
T0.02	0.87	0.71	0.28	0.05	0.56	0.76
T0.05	0.95	0.70	0.29	0.01	0.58	0.78
T0.1	**0.97**	0.57	0.42	**0.004**	0.85	0.68
W2000	0.75	**0.91**	**0.08**	0.21	**0.19**	0.81
W2000d	0.84	0.68	0.31	0.06	0.63	0.73
W3000d	0.82	0.75	0.24	0.07	0.49	0.76

TABLE XIII
EXPERIMENT RESULTS FOR SHORT-TERM EVENTS

	p	*r*	*m*	*f(‰)*	*cost(%)*	*F1*
CD	**0.63**	0.91	0.08	**0.23**	0.19	**0.71**
RD	0.62	**0.92**	**0.08**	0.24	**0.18**	0.70
W2000	0.50	0.91	0.08	0.58	0.22	0.56
W2000d	0.62	0.88	0.11	0.39	0.26	0.67
W3000d	0.61	**0.92**	**0.08**	0.40	0.19	0.68

0.14580), respectively, also reflect event life patterns. As mentioned previously, the magnitude of α multiplied by β determines the speed of decay. The relatively small multiplication of α and β for the sports category shows that sports events are transient compared to political and entertainment events.

E. Comparisons with Other Methods

The focus of the above experiments is to examine whether there exist profiles to describe event aging behavior, and whether event detection can be improved if we utilize the discovered profiles. In this section, the proposed CD scheme and RD scheme are compared with the following three methods: 1) the baseline method (B) [4], which is a basic single-pass clustering algorithm; 2) the time-based threshold method (T) [3]; and 3) the time-window method (W) [30]. Both T and W methods enhance the single-pass clustering algorithm with temporal information. In the time-based threshold approach, each detected event has a detection threshold that determines whether a document is similar enough to be a member of the cluster. Additionally, the threshold is increased periodically by a predefined value so that it is more difficult to incorporate new documents into old detected events. The following equation is used to increase the threshold, where tp is a time penalty [3] set at 0.02, 0.05, and 0.1 (denoted as $T0.02$, $T0.05$, and $T0.1$). The function $d(j,i)$ returns the number of days between document j and the initial time of an event i

$$\mathrm{threshold}(e_i, d_j) = 0.4 + \mathrm{tp} \cdot d(j,i). \tag{26}$$

The Yang *et al.* [30] time-window method limits the detection process to within a window of m previous documents. When processing an incoming document, the similarity between it and every document within the window is computed. A new event is discovered if all similarities fall below a predefined threshold. Otherwise, the document is considered to belong to the event of the most similar document within the window. Yang's decaying-weight time-window method, unlike the simple time-window method that treats all documents in the window equally, gives nearby documents more influence than those farther apart. The following equation defines the decaying-weight similarity between the current document d and the jth document in the time window:

$$\mathrm{decaying\_sim}(d, d_j) = j/m \cdot \mathrm{sim}(d, d_j). \tag{27}$$

Tables X–XII show the comparison results in which $W2000$, $W2000d$, and $W3000d$ are the time-window methods with window sizes of 2000 and 3000, and the lower case d indicates that the window is decaying based. Generally, all temporal-based methods outperform the baseline method. The CD scheme achieves the lowest cost and the best $F1$ score for politics, the second lowest cost and the third $F1$ score for sports, and the second lowest cost and the second $F1$ score for entertainment. Meanwhile, the RD scheme achieves the third lowest cost and the second $F1$ score for politics, the lowest cost and the fourth $F1$ score for sports, and the third lowest cost and best $F1$ score for entertainment. The excellent $F1$ performance indicates that our methods result in well-balanced precision and recall, and the low cost signifies that our methods rarely fail. According to these tables, the time-based threshold method generally has good precision but poor recall, especially when detecting political and entertainment events. The time-window method often achieves high recall, but has a side effect on precision, especially for sports events. In other words, they are not suitable for all kinds of sequential events.

Generally, the fixed window size of the time-window method overemphasizes the influence of documents within the window so that many context similar but event-different documents will be merged into a single event. This shortcoming, which results in low-precision scores, is more noticeable when detecting short-term events, as shown in Table XIII. In addition, the last peak of the curve of the window2000 method shown in Fig. 5 is an example of such mismerging. From these results, it is clear that the time-window method performs poorly when detecting

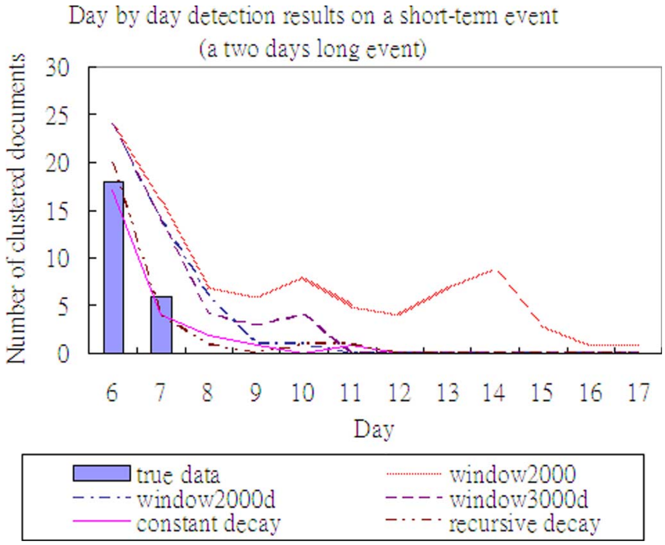

Fig. 5. Event detection results for a short-term event.

TABLE XIV
EXPERIMENT RESULTS FOR LONG-RUNNING EVENTS

	p	*r*	*m*	*f(‰)*	*cost(%)*	*F1*
CD	0.82	**0.71**	**0.28**	0.11	**0.59**	**0.75**
RD	0.87	0.69	0.30	0.08	0.62	0.74
T0.1	**0.93**	0.40	0.59	**0.01**	1.19	0.53
T0.05	0.92	0.51	0.48	0.03	0.98	0.62
T0.02	0.89	0.53	0.46	0.04	0.93	0.63

sports events. Since most sports events are short term, the poor performance of the time-window method can be anticipated. In contrast, our aging methods allow an event to control its own life span; consequently, it outperforms the time-window method for short-term events.

The time-based threshold method does not perform well for political and entertainment events because most of these events are long running. As Table XIV shows, the time-based threshold method accurately groups similar documents together (which achieves high precision), but it cannot detect the complete storylines of long-running events (which results in low recall). Even though the increasing threshold of the time-based threshold method may stay somewhat context-similar, event-different, documents from being included in a detected event, it may break the storylines of long-running events into pieces, which results in a high miss rate. For example, the long-running event shown in Fig. 6 is fragmented into twelve clusters by the time-based threshold method, while our aging methods more accurately break the event into one major and two or three trivial parts.

The results show that the methods used for comparison have shortcomings in detecting sequential events. The time-window method scores high for long-running events, but performs poorly for short-term events, while the time-based threshold has a high-precision rate as well as a high miss rate for long-running events. By comparison, our aging method, which organizes the life cycle of events adaptively, achieves a fairly good performance for both conditions.

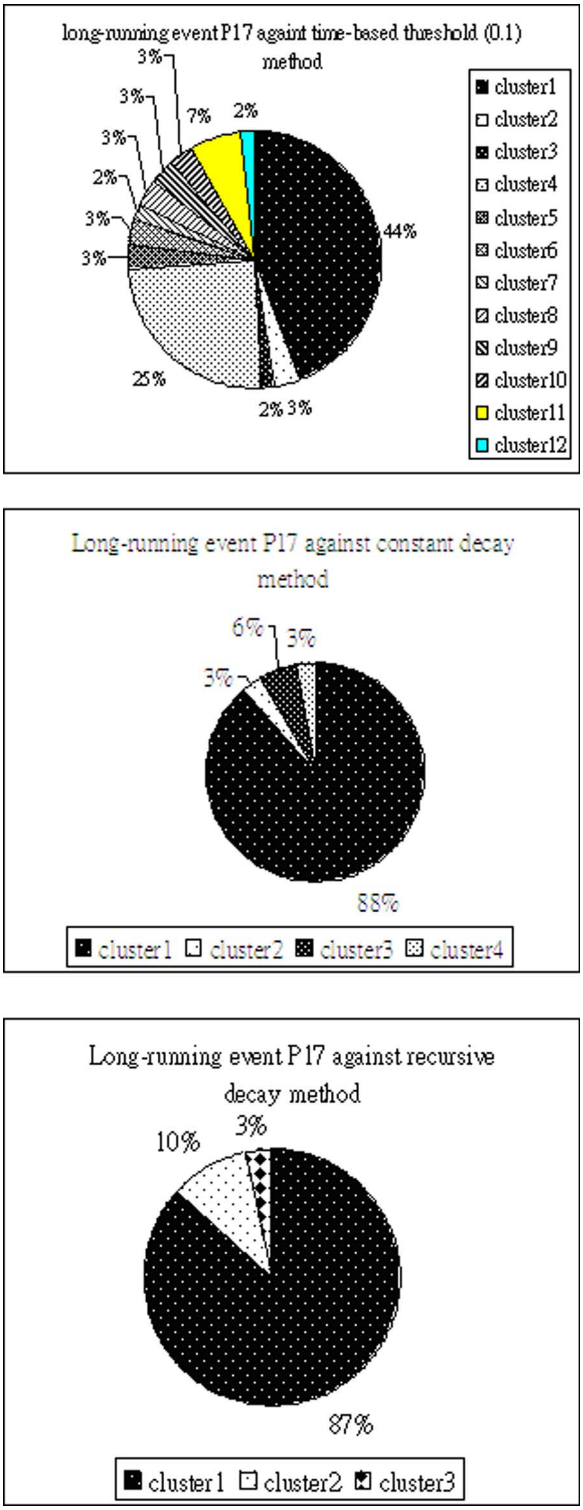

Fig. 6. Event detection results for a long-running event.

F. Evaluation Results of TDT Corpus

In addition to our data corpus, we also evaluate the detection performance using the TDT pilot study corpus [4]. The TDT corpus consists of 15 863 news documents for the period July 1, 1994 to June 30, 1995. Twenty-five labeled events of 1132 documents in the corpus are selected for evaluation. Similar to the evaluation method in the previous section, we

TABLE XV
EXPERIMENT RESULTS FOR THE TDT CORPUS

	p	*r*	*m*	*f(%)*	*cost(%)*	*F1*
CD	0.91	0.54	0.46	0.03	0.90	0.69
RD	**0.94**	0.55	0.45	**0.01**	0.90	0.70
Yang., et al [30]	0.82	**0.62**	**0.38**	0.04	**0.04**	**0.71**
Allan., et al [3]	0.53	0.34	0.66	0.09	0.10	0.42

TABLE XVI
EXPERIMENT RESULTS FOR THE CATEGORIZED TDT CORPUS

	p	*r*	*m*	*f(%)*	*cost(%)*	*F1*
CD	0.88	**0.65**	**0.34**	**0.03**	0.70	**0.75**
RD	**0.90**	0.63	0.36	0.03	0.70	0.74
Yang., et al [30]	0.82	0.62	0.38	0.04	**0.04**	0.71
Allan., et al [3]	0.53	0.34	0.66	0.09	0.10	0.42

employ a cross-validation approach to induce credible results. We first divide the 25 labeled events into five sets, each of which contains five nonoverlapping events. Then, in each cross-validation run, we select one set for testing, and the remaining sets are used for training to obtain the aging parameters. Table XV details the evaluation results.

The poorer performance of our methods on the recall, miss, cost, and F1 measures compared to Yang's approach is due to the variation in life spans of the 25 labeled events. Of the 25 events, seven disappeared within ten days, but 11 lasted over four months. This mixture of long-running and short-term events makes it difficult for the learned aging parameters to describe profile-specific aging behavior. For example, the aging parameters (α, β) of the CD method for the two-day event "Cessna on White House" and the 283-day event "Aldrich Ames" are (0.089, 0.15) and (0.105, 0.0017), respectively. According to (9), a supporting document of an event can help the event survive (the_support_of_the_document $* \alpha)/\beta$ days. Normally, a document contributes 0.3 support value to its related event. Hence, a supporting document of the above two-day event only sustains the event for 0.17 days, but for the long-running event, any one of its supporting documents can sustain the event for more than 62 days. This huge variation indicates that our learned aging parameters perform well on short-term events, but result in poor recall rates for long-running events. Nevertheless, our methods can still maintain high-precision rates and therefore have low false alarms.

According to Table XV, our method's performance is on par with that of Yang when the TDT corpus is used without any precategorization of the training and testing data. Since the TDT corpus does not provide the information for categorization of news documents, we cannot generate aging profiles as in the previous experiments. However, to verify that different profiles have different aging behavior, we divide the 25 labeled events into two classes of similar life spans. The first profile represents long-running events and contains events whose life span is longer than 100 days. The rest of the labeled events form the second profile and are designated as short-term events. As with the previous experiment, we run a cross-validation evaluation on each profile, the detection results of which are listed in Table XVI. The results show that the proposed aging-theory methods achieve better performances than the other approaches when profile-specific training and testing data are used.

The above experiments demonstrate that aging theory is better than other approaches when using profile-specific parameters. But when the training documents are not segmented or categorized, the proposed approach is not better than Yang's method, since the acquired aging parameters cannot precisely describe the aging behaviors as the training documents contain various aging templates.

VI. CONCLUSION

Modeling the life cycles of sequential events is crucial in event detection and tracking. Without a proper life-cycle model, an event may be unnecessarily prolonged by merging similar documents of different events or shortened by rejecting follow-up documents of the same events. In this paper, we have proposed an aging theory that models the life cycle of an event, and incorporate it into a traditional single-pass clustering algorithm to adaptively detect and track online sequential events. Compared with other approaches, our method achieves a fairly good performance for both long-running and short-term events.

The experiments in this paper not only demonstrate the performance of the proposed aging theory, but also confirm our argument that the aging behavior of events is profile dependent. In other words, the event tracking and detection process performs better when the proper aging parameters, trained from training data of the same profile, are used. Even though this observation is quite intuitive, it raises interesting research issues for temporal data mining. The profile mentioned in this paper is not necessarily the narrow definition of category used in a conventional classification system. Instead, a profile can be a group of documents that together form a specific temporal relation. How to define a profile and its criteria for their problem contexts will be an interesting future research topic.

REFERENCES

[1] *Topic Detection and Tracking (TDT) Home.* [Online]. Available: http://www.nist.gov/speech/tests/tdt/index.htm

[2] "JCL—The Java Constraints Library," Artif. Intell. Lab., EPFL. [Online]. Available: http://liawww.epfl.ch/JCL/

[3] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. and Develop. Inf. Retrieval*, 1998, pp. 37–45.

[4] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study: Final report," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 194–218.

[5] J. Allan, V. Lavrenko, and H. Jin, "First story detection in TDT is hard," in *Proc. 9th Int. Conf. Inf. and Knowledge Manage.*, 2000, pp. 374–381.

[6] D. Billsus and M. J. Pazzani, "A personal news agent that talks, learns and explains," in *Proc. 3rd Int. Conf. Auton. Agents*, 1999, pp. 268–275.

[7] R. D. Brown, "Dynamic stopwording for story link detection," in *Proc. Human Language Technol. Conf.*, 2002, pp. 190–193.

[8] C. C. Chen, M. C. Chen, and Y. Sun, "PVA: A self-adaptive personal view agent," *J. Intell. Inf. Syst.*, vol. 18, no. 2/3, pp. 173–194, 2002.

[9] C. C. Chen, Y. T. Chen, Y. Sun, and M. C. Chen, "Life cycle modeling of news events using aging theory," in *Proc. 14th Eur. Conf. Mach. Learning*, 2003, pp. 47–59.

[10] W. B. Frakes and R. Baeza-Yates, *Information Retrieval, Data Structures and Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[11] M. Franz, T. Ward, J. S. McCarley, and W. J. Zhu, "Unsupervised and supervised clustering for topic tracking," in *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. and Develop. Inf. Retrieval*, 2001, pp. 310–317.

[12] R. P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, 4th ed. Reading, MA: Addison-Wesley, 1998.

[13] J. Kleinberg, "Bursty and hierarchical structure in streams," in *Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2002, pp. 91–101.

[14] R. R. Korfhage, *Information Storage and Retrieval*. New York: Wiley, 1997, ch. 6.

[15] V. Kumar, "Algorithms for constraint satisfaction problems: A survey," *AI Mag.*, vol. 13, no. 1, pp. 32–44, 1992.

[16] K. Lang, "NewsWeeder: Learning to filter netnews," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 331–339.

[17] A. Leuski and J. Allan, "Improving realism of topic tracking evaluation," in *Proc. 25th Annu. Int. ACM SIGIR Conf. Res. and Develop. Inf. Retrieval*, 2002, pp. 89–96.

[18] S. H. Lin, M. C. Chen, J. M. Ho, and Y. M. Huang, "ACIRD : Intelligent internet documents organization and retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 3, pp. 599–614, May/Jun. 2002.

[19] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi, "Topic detection and tracking with spatio-temporal evidence," in *Proc. 25th ECIR*, 2003, pp. 251–265.

[20] N. Maria and M. J. Silva, "Theme-based retrieval of Web news," in *Proc. 23rd Annu. Int. ACM SIGIR Conf. Res. and Develop. Inf. Retrieval*, 2000, pp. 354–356.

[21] F. Menczer, R. K. Belew, and W. Willuhn, "Artificial life applied to adaptive information agents," in *Proc. Spring Symp. Inf. Gathering from Distributed, Heterogeneous Database*, 1995, pp. 128–132.

[22] T. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[23] R. Papka, "On-line new event detection, clustering, and tracking," Ph.D. dissertation, Univ. Massachusetts–Amherst, Amherst, 1999.

[24] J. J. Rocchio, "Relevance feedback in information retrieval," in *SMART Retrieval System*. Englewood Cliffs, NJ: Prentice-Hall, 1971, pp. 313–323.

[25] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Reading, MA: Addison-Wesley, 1989.

[26] A. Selamat, H. Yanagimoto, and S. Omatu, "Web news classification using neural networks based on PCA," in *Proc. 41st SICE Annu. Conf.*, 2002, pp. 2389–2394.

[27] D. A. Smith, "Detecting and browsing events in unstructured text," in *Proc. 25th Annu. Int. ACM SIGIR Conf. Res. and Develop. Inf. Retrieval*, 2002, pp. 73–80.

[28] A. S. Tanenbaum, *Computer Network*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[29] H. Wu, T. H. Phang, B. Liu, and X. Li, "Text classification: A refinement approach to handling model misfit in text categorization," in *Proc. 8th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, 2002, pp. 207–216.

[30] Y. Yang, T. Pierce, and J. Carbonell, "A study on retrospective and on-line event detection," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. and Develop. Inf. Retrieval*, 1998, pp. 28–36.

[31] Y. Yang, T. Ault, T. Pierce, and C. W. Lattimer, "Improving text categorization methods for event tracking," in *Proc. 23rd Annu. Int. ACM SIGIR Conf. Res. and Develop. Inf. Retrieval*, 2000, pp. 65–72.

Chien Chin Chen received the B.S. and M.S. degrees in computer science and information engineering from National Central University, Jhongli, Taiwan, R.O.C., in 1997 and 1999, respectively. He is currently working toward the Ph.D. degree in electrical engineering at the National Taiwan University, Taipei, Taiwan.

In 1999, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C., as a Research Assistant. Three years later, he began the Ph.D. program. His research interests include information retrieval, data mining, and machine learning.

Yao-Tsung Chen received the B.B.A. degree in information management, the M.S. degree in computer and information engineering, and the Ph.D. degree in information management from the National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C., in 1995, 1997, and 2001, respectively.

He was a Postdoctoral Fellow with the Institute of Information Science, Academia Sinica, Taipei, Taiwan. He is currently with the Department of Computer Science and Information Engineering, National Penghu University, Penghu, Taiwan. His current research interests include machine learning, text mining, and dynamics modeling.

Meng Chang Chen received the B.S. and M.S. degrees from the National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., and the Ph.D. degree from the University of California, Los Angeles, in 1979, 1981, and 1989, respectively, all in computer science.

He joined the AT&T Bell Labs in 1989, and was an Associate Professor with the Department of Information Management, National Sun Yat-Sen University, Taiwan, from 1992 to 1993. Since then, he has been with the Institute of Information Science, Academia Sinica, Taipei, Taiwan. He has held an Associate Research Fellowship since July 1996. From 1999 to 2002, he took additional responsibility as Deputy Director with the institute. From 2000 to the end of 2003, he served as the Chair of Standards and Technology Transfer group of the National Science and Technology Program for Telecommunications Office (NTPO). His current research interests include data and knowledge engineering, knowledge discovery, networking with QoS supports, multimedia systems and transmissions, and operating systems.