# Life Cycle Modeling of News Events Using Aging Theory*

Chien Chin Chen[1], Yao-Tsung Chen[1], Yeali Sun[2], and Meng Chang Chen[1]

[1] Institute of Information Science, Academia Sinica, Taiwan
{paton,ytchen,mcc}@iis.sinica.edu.tw

[2] Dept. of Information Management, National Taiwan University, Taiwan
sunny@im.ntu.edu.tw

**Abstract.** In this paper, an adaptive news event detection method is proposed. We consider a news event as a life form and propose an aging theory to model its life span. A news event becomes popular with a burst of news reports, and it fades away with time. We incorporate the proposed aging theory into the traditional single-pass clustering algorithm to model life spans of news events. Experiment results show that the proposed method has fairly good performance for both long-running and short-term events compared to other approaches.

## 1 Introduction

Nowadays, the Web has become a huge information treasure. Via the simple Hyper Text Markup Language (HTML) [1], people can publish and share valuable knowledge conveniently and easily. However, as the number of Web documents increases, obtaining desired information from the Web becomes time-consuming and sometimes requires specific knowledge to make best use of search engines and returned results. On-line news reflects such an information explosion problem. It is difficult to access and assimilate desired information from the hundreds of news documents from different agencies generated per day. Techniques such as classification [7][9] and personalization [5][6], were invented to facilitate news reading. However, the classification method is not totally effective in that readers generally follow news by interesting threads, not categories. Moreover, unexpected events, such as accidents, awards and sport championships, are out of the learned user profile. Therefore, to reduce search time and search results a precise event detection method, which discovers news events automatically, is necessary.

Event detection is part of Topic Detection and Tracking (TDT) [2] in which a news event is defined as incidents that occur at some place and time associated with some specific actions. In contrast with a category in the traditional text classification, events are localized in space and time. The job of event detection is to find out new events in several news streams. Besides discussing the TDT techniques of on-line news, in this paper we also discuss one interesting issue about news events — the

* This research was partly supported by NSC under grant NSC 91-2213-E-001-019.

N. Lavrač et al. (Eds.): ECML 2003, LNAI 2837, pp. 47–59, 2003.

event life cycle. Usually, new news events appear in a news burst and gradually die out as time goes on [8]. Ignoring temporal relations of news events will degrade the performance of a TDT system. Previous works [3][14] were aware of the importance of the temporal information of news events to TDT. Their experimental results showed that modeling temporal information of news events could discriminate between similar but distinct events efficiently. In this paper, we propose the concept of aging theory to model life cycles of news events. Experiments show that our approach can improve the deficiencies of other methods.

The rest of the paper is organized as follows. In Section 2, we give a review of related works. In Section 3, we propose the concept of aging theory. Section 4 describes the algorithms that apply the aging theory to a news reading system. We evaluate the system performance in Section 5. Finally, conclusions and future work are given in Section 6.

## 2 Related Works

The project Topic Detection and Tracking (TDT) [2] is a DARPA-sponsored activity to detect and track news events from streams of broadcast news stories. It consists of three major tasks: segmentation, detection and tracking. Our focus, retrospective detection task [3][14], is unsupervised learning oriented [11]. Without giving any labeled training examples, the job of retrospective detection is to identify events from a news corpus. The traditional hierarchical agglomerative clustering (HAC) algorithm [13] is suitable for retrospective detection. However, the computation cost of HAC, which is quadratic to the number of input documents when using group average clustering [14], makes it infeasible when the number of news documents per day is high. Yang, et al. [14] used the technique of bucketing and re-clustering to speed up HAC. However, there is a chance that information from a long running event would be spread over too many buckets and thus divide the event into several events [14].

Another popular approach to retrospective detection is single-pass clustering (or incremental clustering) [4]. The single-pass clustering method processes the input documents iteratively and chronologically. A news document is merged with the most similar detected-cluster if the similarity between them is above a pre-defined threshold; otherwise, the document is treated as the seed of a new cluster. However, by only considering the similarity between clusters and documents will lead context-similar, but event-different, stories to be merged together. In order to obtain better clusters, temporal relations between news documents (or clusters) must be incorporated into the clustering algorithm. Allan, et al. [3] proposed a time-based threshold approach to model the temporal relation. By increasingly raising the detection threshold, distant documents are difficult to align with existing clusters. Therefore, different events could be discerned. Yang, et al. [14] modeled the temporal relation in a time window and a decaying function. The size of a time window specifies the number of prior documents (or events) to be considered when clustering. The decaying function weights the influence of a document in the window based on the gap between it and the examined document. Similar to the time-based threshold approach, distant documents in the time window make less impact on clustering than those nearby.

Even though the above methods enhance the result of the single-pass clustering algorithm, they are not adaptable for all types of event detections. The increasing threshold of time-based threshold method keeps distant stories of long-running events from tracking while the large window size of the time window method may mix up many expired, context-similar, short-term events. In order to balance the tradeoff, and tackle both long-running and short-term events, a self-adaptive event life cycle management mechanism is necessary. We present an aging theory for event cycle in Section 3. For more information about TDT, [4] gives a detailed survey of existing systems and approaches in recent years.

## 3 Aging Theory

A news event is considered a life form with stages of birth, growth, decay and death. To track life cycles of events, we use the concept of energy function. Like the endogenous fitness of an artificial life agent [10], the value of energy function indicates the liveliness of a news event in its life span. The energy of an event increases when the event becomes popular, and it diminishes with time. Therefore, a function of the number of news documents can be used to model the growing stage of events. On the other hand, to model the process of diminishing or aging stages, a decay factor is required.

### 3.1 Notations and Definitions

The news documents to an event is analogous to *foods* to a life form, As various foods do not contribute the same *nutrition* to a life form, different news documents make different contributions to an event's liveliness (i.e. popularity). The degree of the similarity between a news document and an event is used to represent the nutrition contribution. The accumulated similarity between news documents and event $V$ in a time slot $t$ is denoted by $x_t$. The time slot $t$ can be any time interval. In the implementation, we use one day as a time slot.

We then define $\alpha$ as the *nutrition transferred factor* and $\beta$ as the *nutrition decayed factor*, $0<\alpha<1$, $0<\beta<1$, and $y_t = g(x_1,...,x_t, \alpha, \beta)$. $\alpha$ decides the increase of nutrition from an input news document and $\beta$ decides the nutrition loss in a period. The *net nutrition* $y_t$ is a compound variable consisting of the nutrition of each time slot $x_1,...,x_t$, $\alpha$ and $\beta$. Different $g()$s mean different efficiencies of nutrition for different events. A function $F(y)$ has the following properties and is called the energy function of $V$:

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

$F(y)$ is a strictly increasing function of $y$

$$F(\infty) = 1 \text{ and } F(0) = 0 \tag{2}$$

The problem of event life cycle management is to find the optimal combination of $\alpha$ and $\beta$ such that the energy value is 1 when all news documents of the event $V$ appear. However, by Equation (2), the energy value would never be 1. Therefore, we loosen the equation and redefine the optimal condition as

$$F(r \cdot y_T) = s \tag{3}$$

where

$r$ is a proportion of $\Sigma_{i=1,\dots,T}(\alpha x_i)$;
$s$ is a constant;
$T$ is the number of time slots.
Both $r$ and $s$ are selected by the users.

## 3.2 Growth Only

One extreme case of the event life cycle is *no decay*, which means the energy of the event will be accumulated with the clustering of related news documents. In this case, $y_t$ is simply the count of related news documents. Formally, we let $y_t = \Sigma_{i=1,\dots,t}(\alpha x_i)$. We want Equation (3) to hold, so

$$F(r \cdot \sum\nolimits_{i=1,\dots,T}(\alpha x_i)) = s \tag{4}$$

Since the $F$ is a strictly increasing function of $y$, we can take the inverse function $F^{1}$ for both sides of Equation (4):

$$r\sum\nolimits_{i=1,\dots,T}(\alpha x_i) = F^{-1}(s)$$

We then divide both sides by $r\Sigma_{i=1,\dots,T}(\alpha x_i)$ to solve $\alpha$,

$$\alpha^* = F^{-1}(s) / r\sum\nolimits_{i=1,\dots,T}(\alpha x_i) \tag{5}$$

## 3.3 Constant Decay

The extreme case described above is not very likely in real world systems because the energy of an event should not only grow but also eventually diminish with age. Hence, we present the constant decay method which subtracts a constant value for each time slot in this section.

Formally, the domain of the energy function of constant decay is defined as follows:

$$y_t = \sum\nolimits_{i=1,\dots,t}(\alpha x_i - \beta) = \alpha\sum\nolimits_{i=1,\dots,t}(x_i) - \beta t \tag{6}$$

There are two parameters in Equation (6); hence we need two equations to solve them. Let $r_1$, $r_2$ denote two proportions of $\Sigma_{i=1,\dots,T}(\alpha x_i)$. Then

$$y_{t_1} = \alpha r_1\sum\nolimits_{i=1,\dots,T}(x_i) - \beta t_1 \tag{7}$$

and

$$y_{t_2} = \alpha r_2\sum\nolimits_{i=1,\dots,T}(x_i) - \beta t_2 \tag{8}$$

As in the previous section, we substitute Equations (7) and (8) into the optimal condition (3), take the inverse function $F^{1}$ of both sides and get

$$\alpha r_1 \sum_{i=1,\ldots,T} (x_i) - \beta t_1 = F^{-1}(s_1) \tag{9}$$

and

$$\alpha r_2 \sum_{i=1,\ldots,T} (x_i) - \beta t_2 = F^{-1}(s_2) \tag{10}$$

Solve $\alpha$ and $\beta$ by (9) and (10):

$$\alpha^* = [t_2 F^{-1}(s_1) - t_1 F^{-1}(s_2)]/[(r_1 t_2 - r_2 t_1) \sum_{i=1,\ldots,T} (x_i)] \tag{11}$$

and

$$\beta^* = \{r_1[t_2 F^{-1}(s_1) - t_1 F^{-1}(s_2)]/(r_1 t_2 - r_2 t_1) - F^{-1}(s_1)\}/t_1 \tag{12}$$

# 4 Representations and Algorithms

## 4.1 News Document Representation

Both news documents and events were represented as vectors in the conventional vector space model (VSM) [13], but each has a different scheme to determine term weights. For news documents, we use the traditional TF-IDF [13] scheme for term weighting, which is defined as

$$w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t} \tag{13}$$

where

$w_{t,d}$ is the weight of term *t* in document *d*;

$tf_{t,d}$ is the within-document term frequency (TF);

$log(N/df_t)$ is the inverted document frequency (IDF),

*N* is the number of documents in the system corpus,

$df_t$ is the number of documents in the corpus which *t* occurs.

The term weights of a news event are obtained from a set of detected documents. However, due to the temporal relation of news documents, the event's weights must be updated progressively to reflect event evolution. We adopt the classic Rocchio method [12] to update the term weights of events incrementally.

$$w_{t,e} = (1 - \gamma) \cdot w_{t,e} + \gamma \cdot w_{t,d} \tag{14}$$

where

$w_{t,e}$ is the weight of term *t* in the detected event *e*;

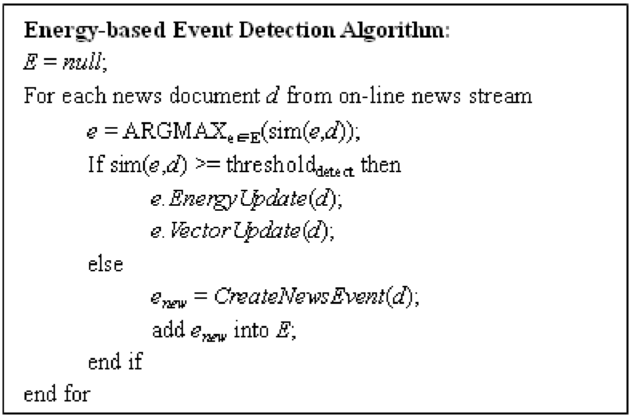
$w_{t,d}$ is the weight of term *t* in the inserted document *d*;

$\gamma$ is a parameter between 0 and 1.

Simply, the term weight of an event is a weighted combination of its original term weight and the term weights of newly detected news documents. Besides the term vector, we also assign each event a real number *eng*, called *energy value*, to indicate its vitality. The energy of an event increases when the event is popular, and it decreases with a constant value for each period of time. Therefore, events that receive little interest will gradually fade out.

### 4.2 Event Detection Algorithm

Based on the aging theory described in Section 3, the energy-based event detection algorithm is as in Figure 1. *E* is a set of candidate news events which are detected by the algorithm. Initially, *E* is set as empty. For each incoming news document *d*, the similarity between *d* and the most similar detected event in *E* is examined against a predefined threshold, called $threshold_{detect}$. The similarity is the cosine function between the vectors of news document and news event.



```
Energy-based Event Detection Algorithm:
$E = null$;
For each news document $d$ from on-line news stream
        $e = \text{ARGMAX}_{e \in E}(\text{sim}(e,d))$;
        If $\text{sim}(e,d) >= \text{threshold}_{detect}$ then
                $e.EnergyUpdate(d)$;
                $e.VectorUpdate(d)$;
        else
                $e_{new} = CreateNewsEvent(d)$;
                add $e_{new}$ into $E$;
        end if
end for
```

**Fig. 1.** The energy-based event detection algorithm.

If the similarity is greater than the threshold, we classify the news into the event and update the event's term vector and energy value. *EnergyUpdate*() increases the energy value of the corresponding event. Intuitively, the similarity values between the event and its documents can be summed as the energy value. However, this unbounded value will cause hot event, consisting of a burst of news documents, to have huge energy values and a longer life span than the event actually has. To overcome this pitfall, the energy function defined in Section 3 is used to constrain the energy value. The formula of *e.EnergyUpdate*(*d*) is defined as

$$eng_e = F(F^{-1}(eng_e) + \alpha \cdot sim(e,d)) \tag{15}$$

where

$eng_e$ is the energy value of the event *e*;
*F*() is the energy function;
$F^{-1}$() is the inverse energy function;
*sim(e,d)* is the cosine value between the event *e* and the document *d*;
$\alpha$ is the energy transferred factor.

In this study, we adopt a sigmoid function as the energy function which converts the sum of similarities into a bounded extent. The sigmoid function is defined as:

$$F(y)=\frac{10y}{1+10y}\text{, y>0} \qquad (16)$$

=0, otherwise.

One of the distinguishing features of the sigmoid function is that it maps a very large input domain into a small output domain. As shown in Figure 2, the output ranges between 0 and 1. Therefore, by using the sigmoid function, energy values can be limited between 0 and 1, which is consistent to the definition of the energy function described in Section 3. Another interesting feature of the sigmoid function is that it is a nonlinear function. The curve is much steeper around the origin than the extremities. This kind of growth often reflects the development of an event. It is usually accompanied by a burst of news documents in the beginning, and then gradually fades away. Since the energy value is constrained, we can interpret the status of the event by partitioning the range of the output of the sigmoid function. In practice, we can divide the output of the sigmoid function into three parts, each part representing a different situation of an event. High sigmoid values indicate hot events, and low sigmoid values indicate that events are out of date.

After increasing the energy value, the function *VectorUpdate()* is called to capture up-to-date event status. We use the above Rocchio formula [12] to update the term vector of the detected event. That is, we assume the inserted document *d* is positive feedback and use it to adjust the term weights of the event.

In case the similarity between the incoming document *d* and the most similar event is smaller than $threshold_{detect}$ or the set of *E* is empty, the detection algorithm will create a new event. The function *CreateNewsEvent*() forms the term vector of the newly created event by copying it from the input news document.

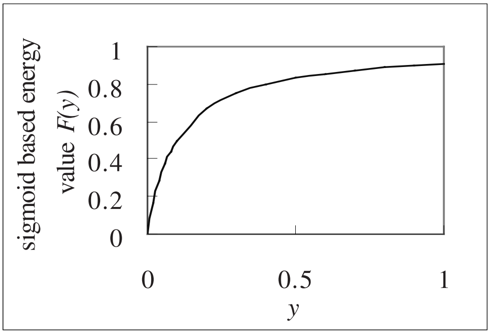


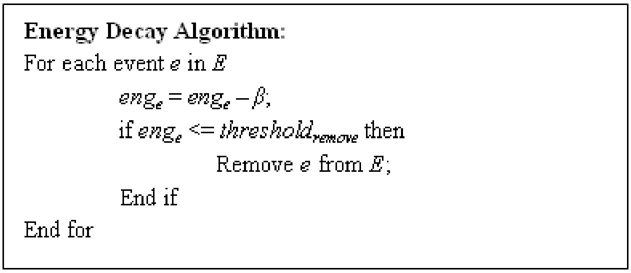
**Fig. 2.** The graph of the sigmoid function.

### 4.3 Energy Decay Algorithm

In contrast to the energy-based event detection algorithm which generates new events, the energy decay algorithm shown in Figure 3 is used to remove antiquated events.

Since events are time dependent, an event detection system is defective unless it can remove expired events. The energy decay algorithm periodically (e.g., every midnight) checks the energy values of events and removes antiquated events to keep all events in a news reading system up to date.

The energy value of every detected event is periodically reduced with a decay factor $\beta$, and the value of $\beta$ is calculated by the aging theory. When no or few documents are added to an event, its energy value will gradually decline. Moreover, if an event's energy value is lower than a predefined threshold, called $threshold_{remove}$, we suppose the event is out of date and remove it from the validated set *E*. With the energy-based event detection algorithm and the energy decay algorithm, the lifespan of a news event can be determined by the liveliness of the event. The more related news documents it has, the longer its lifespan. This makes life cycle management of news events self-adaptive.



```
Energy Decay Algorithm:
For each event $e$ in $E$
        $eng_e = eng_e - \beta$;
        if $eng_e <= threshold_{remove}$ then
                Remove $e$ from $E$;
        End if
End for
```

**Fig. 3.** The energy decay algorithm.

## 5 Empirical Evaluations

Two experiments were designed to evaluate and verify the proposed theory. In the first experiment, the training part of our data corpus is used to acquire the optimal aging parameters. We use the learned parameters to plot the variation of energy values of events in the testing part of the corpus. Some interesting observations from this experiment are discussed below. Then, we evaluate the performance of our method against three other methods in terms of traditional TDT metrics. The experimental results show that our method is more adaptive than others in both long-running and short-term events.

### 5.1 Data Corpus

Table 1 details the corpus we made by collecting news documents from several online news agencies for evaluations. We forgo the TDT pilot study corpus [4] for evaluations because it does not offer us a set of training data to obtain the aging parameters. Moreover, we believe that the aging parameters are category-derived. Each category will have its own best aging parameters in relation to its events. Therefore, we compile a corpus that comprises events of all sorts of categories to conclude our purpose. In this study, 18 events in politics are used for evaluations. In the future, events of other categories will be used as well. Besides, we have categorized types of events based on the event period. Events were identified as short-term events if they vanished within three days. In contrast, if the life of an event lasted over a week, we call the event a long-running event. Categorizing the type of events could help us in discussing the strength of each of the comparing methods in different situations.

**Table 1.** Statistics of data corpus.

| | Training Data | Testing Data |
|---|---|---|
| Start-end date | 2002/10/1 – 2002/10/31 | 2002/11/1 – 2002/12/31 |
| Number of news documents | 13,267 | 30,256 |
| Number of labeled events | 8 | 10 |
| Event period <= 3 | 2 | 1 |
| 7 >= event period > 3 | 2 | 3 |
| **Event period > 7** | 4 | 6 |

### 5.2 Experiment 1: Growth of Energy Values of News Events

This experiment inspects the effect of aging parameters on the life cycle of news events. According to formulas (11) and (12), the values of $\alpha$ and $\beta$ are determined by the points $(r_{1,}s_1)$ and $(r_2,s_2)$. We set $r_1$ and $s_1$ as 0.20 and $r_2$ and $s_2$ as 0.85 and the corresponding $t_1$ and $t_2$ are replaced by the times where the sum of similarity exceeds the $r_1$ and $r_2$ portion of the sum of similarities between the event and all its documents. The final values of $\alpha$ and $\beta$, 0.118659 and 0.145198 respectively, are the averages of the results of all training events.

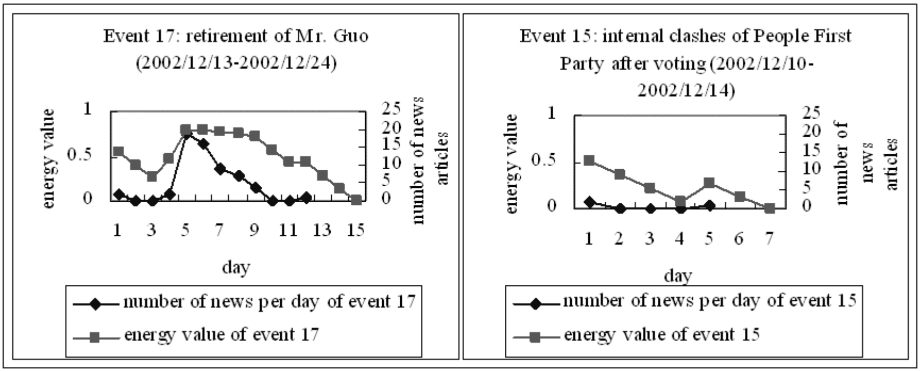


**Fig. 4.** The growth of the energy value.

Figure 4 shows the energy values and numbers of news documents per day of some of the testing events. We chose these events because they are difficult tasks in event detection. The event in the left began with very few related news documents, and it immediately quieted for a while until some follow-up news occurred. During the quiet period, it is hard for an event detection system to identify whether the event has ended, especially if it had a weak beginning. An early death announcement will cause serious errors since there were plenty of follow-ups. However, holding all weak beginning events will overly emphasize the importance of weak events, which may cause false alarms. Fortunately, our aging theory could tackle this kind of dilemma. Since the sigmoid function is a nonlinear function, the steeper slope near the origin point gives an event a higher vitality even if that event is inactive at an early stage. As a result, the high initial energy value helps the event survive the quiet period to track the follow-ups. What if the event is indeed a weak event, as Event 15 in figure 4? In this case, our decay mechanism could eliminate this event after a short period of time. As shown in figure 4, this event only lasted for two more days. Observing the contrast between events in Figure 4, we found that our aging theory could ascertain the rise and fall of an event progress.

## 5.3 Experiment 2: Event Detection Comparisons

In this experiment, our aging method (*A*) is compared to three proposed methods. The baseline method (*B*) [4] is a basic single-pass clustering algorithm. The time-based threshold method (*T*) [3] and the time window method (*W*) [14] enhanced the single-pass clustering algorithm with temporal information. Each of the above methods groups the testing part of the corpus into several clusters. The top ten best-matched clusters generated from each method were chosen for performance comparisons. The degree of match between a testing event and a generated cluster is determined by the number of news documents belonging to both the testing event and the cluster. As a result, 10 generated clusters from each method are evaluated using six official TDT measures [4] including: precision (*p*), recall (*r*), miss (*m*), false alarm (*f*), F1-measure (*F1*) and cost (*c*).

Table 2 shows the results of the experiments in which, *T0.02*, *T0.05*, and *T0.1* are time-based threshold methods with 0.02, 0.05 and 0.1 time penalty [3] respectively. *W2000*, *W2000d*, and *W3000d* are the time window methods with window sizes [14] of 2000 and 3000 respectively. The lowercase letter *d* indicates that the window is decaying-based [14].

**Table 2.** Experiment results on testing events.

| | ***P*** | ***r*** | ***m*** | ***f*** | ***F1*** | ***c*** |
|---|---|---|---|---|---|---|
| ***B*** | 0.68 | 0.75 | 0.25 | 0.0007 | 0.63 | 0.006 |
| ***A*** | 0.73 | 0.75 | 0.24 | 0.0002 | **0.72** | 0.005 |
| ***T0.02*** | 0.88 | 0.58 | 0.41 | 0.00008 | 0.67 | 0.008 |
| ***T0.05*** | 0.87 | 0.57 | 0.42 | 0.00008 | 0.67 | 0.008 |
| ***T0.1*** | **0.93** | 0.47 | 0.52 | **0.00003** | 0.61 | 0.01 |
| ***W2000*** | 0.56 | **0.89** | **0.1** | 0.002 | 0.62 | **0.004** |
| ***W2000d*** | 0.76 | 0.67 | 0.32 | 0.0002 | 0.65 | 0.006 |
| ***W3000d*** | 0.65 | 0.8 | 0.19 | 0.0006 | 0.66 | **0.004** |

In table 2, approximately all temporal-based methods out-perform the baseline method. Our aging method achieves both reasonable precision and recall which results in the best *F1* score, while the time-based threshold method achieves good precision but loses recall. The time window method has good recall but decreased precision. Even though the lowest cost comes from the time window method, we believe that is due to the fact that the majority of testing events are long-running. If we separately compare the time window method with our aging method on a short-term event, as shown in table 3, we find that the time window method is not suitable for short-term events. During the detection process on a short-term event, the fixed window size will overly emphasize the influence of the short-term event so that many context-similar but event-different news stories are merged into the event, which therefore results in low precision scores. As indicated in Figure 5, the last peak of the curve of the *window2000* method is a mis-merged event. Our aging method, on the other hand, lets the event control its own lifespan; consequently it outperforms the time window method for short-term events.

**Table 3.** Experiment results on short-term event.

| | ***P*** | ***r*** | ***m*** | ***F*** | ***F1*** | ***c*** |
|---|---|---|---|---|---|---|
| ***A*** | **0.61** | 0.79 | 0.2 | **0.0003** | **0.69** | **0.004** |
| ***W2000*** | 0.12 | **0.91** | **0.08** | 0.005 | 0.21 | 0.006 |
| ***W2000d*** | 0.37 | 0.83 | 0.16 | 0.001 | 0.51 | **0.004** |
| ***W3000d*** | 0.22 | 0.83 | 0.16 | 0.002 | 0.34 | 0.005 |

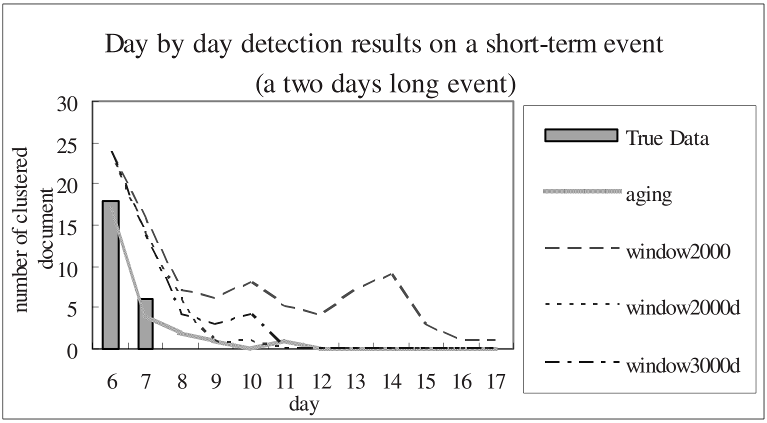


**Fig. 5.** Event detection results on a short-term event.

The time-based threshold method sacrifices its recall to achieve high precision, especially in long-running events (as shown in table 4). The growing threshold could keep somewhat context-similar, but event-different, news stories from being included in a detected event. Thus it results in a substantial amount of clusters. However, continuingly increasing the threshold may break the storylines of a long-running event into pieces, which consequently results in a high miss rate. As we can see from into pieces, which consequently results in a high miss rate. As we can see from Figure 6, the long-running event was fragmented into twelve clusters when using the time-based threshold method, while our aging method merely broke the event into one major and three trivial parts.

**Table 4.** Experiment results on 6 long-running events.

| | *p* | *r* | *m* | *f* | *F1* | *c* |
|---|---|---|---|---|---|---|
| ***A*** | 0.72 | **0.68** | **0.31** | 0.0002 | **0.67** | **0.006** |
| ***T0.02*** | 0.88 | 0.46 | 0.53 | 0.00005 | 0.60 | 0.01 |
| ***T0.05*** | 0.87 | 0.51 | 0.49 | 0.00007 | 0.61 | 0.009 |
| ***T0.1*** | **0.94** | 0.37 | 0.62 | **0.00002** | 0.52 | 0.012 |

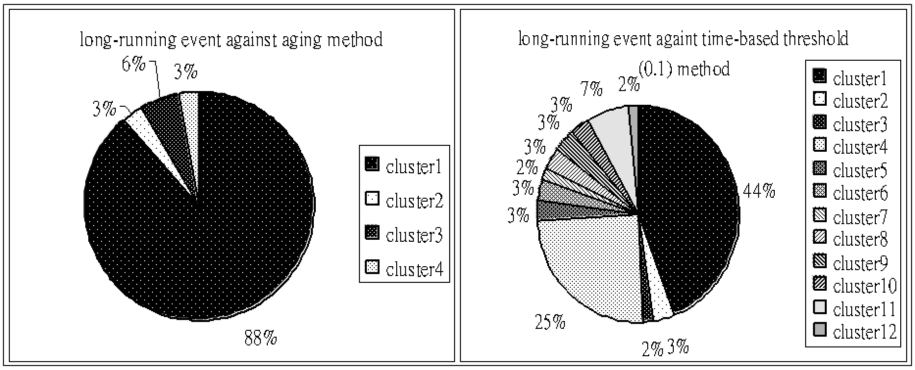


**Fig. 6.** Aging method on a long-running event.

Experiment results show that the time window method scores high on long running events but perform poorly for short-term events. The time-based threshold has a high precision rate as well as a high miss rate on long-running events. Our aging method has fairly good performance in both long-running and short-term events.

## 6 Conclusions

In this study, the aging theory is incorporated into the traditional single-pass clustering algorithm to detect and track news events. The growth of energy value synchronizes well with the event progress. Moreover, experiment results show that the proposed method performs well for both long-running and short-term events in comparison to other approaches. We are now applying our method to other categories, such as finance, sports and entertainment. We believe that using category-specific aging parameters achieves the best results for event detection in all possible categories.

## References

1. http://www.w3.org/MarkUp/
2. http://morph.ldc.upenn.edu/TDT/
3. James Allan, Ron Papka and Victor Lavrenko, "On-Line New Event Detection and Tracking," in proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, pp. 37-45, 1998.
4. James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron and Yiming Yang, "Topic Detection and Tracking Pilot Study: Final Report," in proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 194-218, 1998.
5. Daniel Billsus, Michael J. Pazzani, "A Personal News Agent that Talks, Learns and Explains," Third International Conference on Autonomous Agents, 1999.
6. Chien Chin Chen, Meng Chang Chen and Yeali Sun, "PVA : A Self-Adaptive Personal View Agent," Journal of Intelligent Information Systems, 18:2/3, 173-194, 2002.
7. Wen-Lin Hsu, Sheau-Dong Lang, "Classificaiton Algorithms for NETNEWS Articles," in proceedings of 8th ACM international conference on information and knowledge management, pp. 114-121, 1999.
8. Jon Kleinberg, "Bursty and Hierarchical Structure in Streams," in proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining, pp. 91-101, 2002.
9. Shian-Hua Lin, Meng Chang Chen, Jan-Ming Ho and Yueh-Mn Huang, "ACIRD : Intelligent Internet Documents Organization and Retrieval," IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 3, May/Jun, 2002.
10. Filippo Menczer, Richard K. Belew and Wolfram Willuhn, "Artificial Life Applied to Adaptive Information Agents," In Spring Symposium on Information Gathering from Distributed, Heterogeneous Database, AAAI Press, 1995.
11. Tom Mitchell, "Machine Learning," McGraw Hill, 1997.
12. Joseph John Rocchio, "Relevance Feedback in Information Retrieval," In the SMART Retrieval System, pp. 313-323. Prentice Hall, 1971
13. Gerard Salton, "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer," Addison-Wesley, 1989.
14. Yiming Yang, Tom Pierce and Jaime Carbonell, "A Study on Retrospective and On-Line Event Detection," in proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, pp. 28-36, 1998.