World Wide Web (2008) 11:1–37
DOI 10.1007/s11280-007-0018-9

A Novelty-based Clustering Method for On-line Documents

Sophoin Khy · Yoshiharu Ishikawa · Hiroyuki Kitagawa

Received: 11 December 2006 / Revised: 3 January 2007 /
Accepted: 5 January 2007 / Published online: 17 February 2007

Abstract In this paper, we describe a document clustering method called *novelty-based document clustering*. This method clusters documents based on similarity and *novelty*. The method assigns higher weights to recent documents than old ones and generates clusters with the focus on recent topics. The similarity function is derived probabilistically, extending the conventional cosine measure of the vector space model by incorporating a *document forgetting model* to produce novelty-based clusters. The clustering procedure is a variation of the K-means method. An additional feature of our clustering method is an incremental update facility, which is applied when new documents are incorporated into a document repository. Performance of the clustering method is examined through experiments. Experimental results show the efficiency and effectiveness of our method.

Keywords document clustering · forgetting factor · incremental processing · novelty · on-line documents

S. Khy (✉) · H. Kitagawa
Graduate School of Systems and Information Engineering, University of Tsukuba,
1-1-1 Tennohdai, Tsukuba, Ibaraki 305-8573, Japan
e-mail: sophoin@kde.cs.tsukuba.ac.jp

Y. Ishikawa
Information Technology Center, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
e-mail: ishikawa@itc.nagoya-u.ac.jp

H. Kitagawa
Center for Computational Sciences, University of Tsukuba,
1-1-1 Tennohdai, Tsukuba, Ibaraki 305-8573, Japan
e-mail: kitagawa@cs.tsukuba.ac.jp

1 Introduction

With the rapid development of Internet technology and less expensive hardware, electronic documents like news articles and scientific papers have proliferated and are delivered continuously over time. This explosion of electronic documents has made it difficult for a user to extract useful information from them. This issue has been approached using various techniques, one of which is *document clustering*. Document clustering is used as a core technique in managing vast data and providing summarized information. It collects similar documents into groups. Document clustering has been used as a fundamental method in many areas, such as information retrieval [4, 12, 32, 33], information filtering [14], and topic detection and tracking [1]. It has also been used as a preprocessing step for other document processing tasks, such as text classification [33] and summarization of documents [9, 30, 39] and as an analysis approach in Web information management, including the study of Web communities [40].

In this work, we consider another problem in organizing on-line documents. In an on-line environment, a user tends to be interested in new and up-to-date information, for example, when browsing on-line news. Although traditional document clustering methods can provide clusters of relevant documents to the user to assist the browsing task, they do not fulfill the requirement of a user interested in recent issues since the provided clusters consist of old and new topics. With this background, we propose a *novelty-based document clustering method* [17, 19], which summarizes trends of on-line documents and provides users with up-to-date information. The novelty-based document clustering method works on-line, focusing on recent documents. The objective of the method is to generate clusters reflecting trends of recent topics by presenting up-to-date cluster snapshots.

The novelty-based clustering method is based on a *novelty-based similarity measure*, which is an extension of a traditional approach in information retrieval, the cosine similarity measure in the vector space model. It is also related to the idea of the $tf \cdot idf$ term weighting scheme, but is derived in terms of probabilistic formulation based on the concept of a *document forgetting model*. The clustering algorithm is an extended version of the K-means method, often used in information retrieval. The prominent characteristic of our clustering method is our incorporation of the *forgetting factor* to formulate the similarity metric. We assume that each document has its importance value (called a *weight*). A document weight has an initial value *one* when a document arrives; it then gradually decays according to the forgetting factor. Based on the forgetting factor, the clustering method assigns higher weights to recent documents than older ones. In other words, the method gradually forgets old documents and focuses mainly on recent documents to generate clusters. An additional feature of the clustering method is its incremental update facilities. The incremental update feature is suited to on-line environments where documents are continually delivered.

This paper is partially based on our previous work [17] and [19]. In [17], we proposed the forgetting-factor-based similarity function and derived a clustering algorithm that extends the incremental clustering method by Can [5]. Reference [19] is a preliminary version of this paper. In [19], we used the novelty-based similarity function in [17]; we proposed a variation of the K-means clustering algorithm and performed a preliminary experimental evaluation. This paper extends our previous

work and examines the efficiency and effectiveness of the novelty-based clustering method through more detailed experiments.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the novelty-based similarity measure. The clustering algorithm is presented in Section 4. Section 5 reports the experimental evaluation of the clustering method. Section 6 concludes the paper.

2 Related work

In this section, we overview the processing of time-series documents. Then we review document clustering methods. A description of the notions of obsolescence and temporal decay, based on which we derived the document forgetting model, comes next.

2.1 Processing of time-series documents

The focus of our research is to cluster what we call *time-series documents*, which are documents continually delivered with timestamps, and to summarize recent hot topics. News articles, electronic mail, RSS, weblogs, etc. are examples of time-series documents. Their qualitative aggregation and summarization are important to support user activities in the Web world. Research on processing documents continually delivered in time order is an interesting research area in information retrieval and Web information management and has gained substantial interest. Intensive studies have tried to find meaningful and important information or structures in time-series documents. The following reviews some research work in this area.

A research program relevant to our work is the *topic detection and tracking* (TDT) [1]. It is a research program organized by the National Institute of Standards and Technology (NIST) [29]. It tries to organize on-line documents like broadcast news based on the notions of events and topics. TDT tasks that process time-series documents include topic detection to detect clusters of stories that discuss the same topic; topic tracking to keep track of stories similar to a set of example stories; and new event detection to detect if a story is the first story of a new, unknown topic. In TDT, clustering approaches have been used in some TDT tasks. Research papers related to these tasks include [2, 16, 24, 36, 37] for topic detection and topic tracking tasks, and [22, 34, 38] for new event detection. TDT's topic detection task is closely related to our work. However, our approach not only clusters documents into topics, but also focuses on recent documents to generate clusters of recent topics. We describe a clustering method proposed for the topic detection task in Subsection 2.2.

Chronologically ordered temporal information of time-series documents has been exploited in many different ways and domains. In recent study on weblog, Mei et al. studied the problem of discovering and summarizing the spatiotemporal theme (or subtopic) patterns in weblogs [26]. They proposed a probabilistic approach to modelling the subtopic themes and their distribution and evolution patterns over time and location. In [20], Kleinberg studied the bursts of topics in document streams. To analyze the bursts of a topic in the document stream, he considered document streams as temporal frequency data and modeled the stream using an infinite-state automaton focusing on the arrival rate of documents related to a topic.

The study of bursty evolution of blogspace was conducted by Kumar et al. [21]. They created a time graph on blogspace, a graph that evolves in continuous time, by an automatic analysis of blogs internal timestamps and studied the evolution of connected component structure and microscopic community structure in the time graph and developed algorithms to track evolution of the blog community. They also extended Kleinberg's method [20] to discover bursty communities of blogs that are topically and temporally focused. Cui et al. [11] analyzed topic activation from document streams based on the Kleinberg's method [20], considering document arrival rates, relevance, temporal degradation and incremental schemes. Mei et al. [27] proposed a method to discover and summarize the evolution of thematic (i.e., subtopic) structures in a text stream. News articles and abstracts of research papers were used in their experiments. Our work differs from the existing work. We utilize temporal information of each document to assign a weight to the document. The weight decreases as the clustering evolves over time. In other words, old documents have smaller weights than recent ones.

Our research addresses the issue of managing time-series documents from a perspective different from the above research. The focus of our research is to cluster time-series documents and to summarize the trend of recent hot topics. We incorporate a decaying function in the similarity measure and cluster documents incrementally based on a variant of the *K*-means method.

2.2 Document clustering

So far, numerous clustering approaches have been proposed. In [18], Jain et al. provides a general survey of clustering methods. They classified clustering approaches into wide ranges of categories and reviewed clustering methods in each category. Those conventional approaches generally focus on developing efficient and effective clustering methods, which group similar objects into same clusters. To our knowledge, no approaches with the same objective as our novelty-based clustering approach have appeared as we prepare this paper. Here we review literature in document clustering by selecting clustering methods that are closely related to our approach.

The *K-means clustering* method [18, 25], based on which we devise our algorithm, is one of the most widely used clustering methods. The method is known for its efficiency compared to the hierarchical clustering method. Given n objects, the method first select k objects as initial k clusters. It then iteratively assigns each object to the most similar cluster based on the mean value of the objects in each cluster. There are many variations of the *K*-means method. In our approach, we use the *K*-means method with extensions to cope with incremental processing and outlier handling. Section 4 gives details.

A feature of our clustering method is its incremental processing. Incremental processing is required since the target data of our method is on-line documents that are delivered continually. Updates are needed when new documents are incorporated and when documents are deleted because they become obsolete as clustering targets. Coping with a small number of updates by re-computing the whole clustering from scratch is costly, especially when the document set is very large. There are several proposals of incremental clustering methods. Can proposed a clustering

algorithm called C^2ICM (cover-coefficient incremental clustering methodology) [5], which is an incremental version of C^3M (cover-coefficient-based concept clustering methodology). It is based on the concept of a *cover coefficient*, which measures the degree that a document is covered by other documents and determines the number of clusters and cluster seeds automatically. C^2ICM enhances C^3M, allowing documents to be incrementally added and deleted.

Single-pass clustering can also be classified as an incremental clustering method. In TDT 1998 competition, Yang et al. proposed the use of a single-pass incremental clustering method for retrospective detection and on-line detection of the topic detection task [36, 37]. The method sequentially processes input documents one at a time and maintains clusters incrementally. A new document is assigned to a cluster if the similarity score between the document and the cluster is above a preselected threshold. Otherwise, a new cluster is generated and the document becomes its seed. For on-line detection, the method imposes a time window and incorporates a linear decaying-weight function into the similarity function. Their approaches revealed relatively good performance in the TDT evaluation situation.

Incremental clustering is also proposed in other research areas. Charikar et al. defined an incremental clustering model for the hierarchical agglomerative clustering (HAC) in [7]. However, the approach assumes that the data is located in a metric space, and related notions, such as distances and diameters, are used. Therefore, it is not useful in our context. The same comments can be applied to other incremental clustering methods [8].

In contrast to related work, our approach extends the conventional *K*-means method. The reasons are as follows:

1. In clustering, there is a trade-off between efficiency and quality. The *K*-means clustering method is a commonly used approach and its basic algorithm is quite simple. Compared to other conventional hierarchical methods, the processing cost is lower in general and suited for on-line document clustering. Although the single-pass method is superior to the *K*-means method in terms of processing cost, the clustering quality of the former is usually worse than that of the latter.
2. It can support incremental insertion and deletion of documents efficiently. Some existing clustering methods, which have features of incremental processing (e.g., [41]), do not support efficient deletion of old items. The feature is important in our context since old documents are obsolete and should be excluded from the clustering targets.
3. It can be extended to be robust for *outliers*. As shown later, our similarity function causes the effect that old documents are "forgotten" and no longer considered—they become outliers. Although the existence of outliers adversely affects clustering results, our extension to the *K*-means method can avoid such problems.

The clustering algorithm is detailed in Section 4.

2.3 Obsolescence and temporal decay

The notion of *obsolescence* has been widely studied in library information science and informetrics [15]. Obsolescence is the decrease in the use of documents as they

age. Obsolescence is also called *aging* or *decay* [13]. Previous studies on the aging of documents have highlighted the behavior and tendency of how the citation of a paper or journal rises or falls. For example, in a study on citation analysis of individual papers or journals, Avramescu [3] suggested the following equation for the citation distribution of papers and journals:

$$y(t) = C\left(e^{-\alpha t} - e^{-\beta t}\right), \tag{1}$$

where $\beta > \alpha$ and C is a constant. The parameters α and β are used to model growth and decay of citations, respectively. That is, at the beginning of its publication, the number of citations of the paper or journal is zero. It then increases to a degree depending on the significance of the paper or journal; it reaches maximum and then begins to fall. In other words, the model states that the number of citations rises exponentially and also decreases exponentially. Temporal decay functions such as linear and exponential decay functions are often used in other areas such as clustering [6, 28], TDT research [36, 37], and stream data processing [10].

The similarity measure used in this paper incorporates an exponential aging function derived from the document forgetting model [17], which is presented in Subsection 3.2. The model assumes that, at the beginning, the influence value (weight) of a document takes the highest value *one*. The value then exponentially decreases according to a decay factor value. Adoption of the exponential decay factor in the similarity function used in this paper is derived based on the previously well-studied approach in obsolescence. Beyond that, the exponential decay function enables efficient incremental update, as explained in Subsection 3.5.

3 A novelty-based similarity measure

In this section, we describe the proposed similarity metric incorporating the notion of temporal decay. Firstly, we introduce the document forgetting model, followed by a description of the similarity measure derived from the model. Then we explain the method to set parameter values introduced in our method. As a final item, we show the efficient computation approach for the update of the statistics and probabilities required to compute similarity scores.

3.1 Symbols and definition

Table 1 summarizes the definitions of symbols used in this paper.

3.2 Document forgetting model

The *document forgetting model* plays an important role in our method. The model is based on a simple intuition: on-line documents such as news articles and journal articles maintained in a document repository lose their values gradually as time passes. We propose the following exponential weighting formula to represent decay of a document's influence value.

Table 1 Symbols and their definitions.

Symbol	Definition
d_i	Document in the document set
λ	Forgetting factor
T_i	Acquisition time of d_i
τ	Current time
dw_i	Weight of d_i
$x\|_\tau$	Value of variable x at current time
$\Pr(d_i)$	Selection probability of d_i
tdw	Total weight of documents in the document set
t_k	Index term
$\Pr(t_k\|d_i)$	Probability that t_k is selected from d_i
$\Pr(t_k)$	Occurrence probability of t_k
$\Pr(d_i, d_j)$	Co-occurrence probability of d_i and d_j
$sim(d_i, d_j)$	Similarity score between d_i and d_j
tf_{ik}	Term frequency of t_k in d_i
f_{ik}	Number of occurrences of t_k in d_i
idf_k	Inverse document frequency of t_k
len_i	Document length of d_i
ε	Expiration parameter
β	Half-life span parameter
γ	Life span parameter
G	Clustering index
$avg\_sim(C_p)$	Average similarity of documents in cluster C_p

Definition 1 (Document weight) Let the current time be $t = \tau$ and the acquisition time of each document d_i $(i = 1, \ldots, n)$ be T_i $(T_i \leq \tau)$. We define the *weight* of d_i at time τ, $dw_i|_\tau$, by

$$dw_i|_\tau \stackrel{\text{def}}{=} \lambda^{\tau - T_i} \quad (0 < \lambda < 1), \tag{2}$$

where λ is a parameter tuned according to the target document set and the intended application. We call λ a *forgetting factor*.

Figure 1 depicts the exponential degradation of the document weight. When a document is acquired, its document weight is *one*. As time passes, the weight decreases exponentially. Decay speed is specified by the forgetting factor λ. The smaller the value of λ, the faster the forgetting speed becomes. For the acquisition time of on-line documents, we can use the issue date as the acquisition time. The notation "$|_\tau$" is used to represent the value of a variable at time τ. If the context is clear, we omit "$|_\tau$". We describe an intuitive way of setting the forgetting factor λ's value in Subsection 3.4.

The reasons for selecting this exponential forgetting model are summarized as follows:

1. The document forgetting model is based on the concept of obsolescence in citation analysis. We borrowed the basic idea of "exponential forgetting" from citation analysis literature—a typical one is shown in Eq. 1. However, we simplified it to provide a concise model and efficient implementation. The document forgetting model is based on the intuition that on-line documents such as news

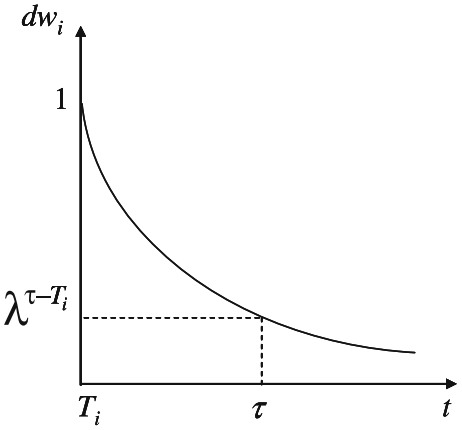

Figure 1 Exponential decay function.

stories receive immediate attention from readers at the beginning, but their interest falls gradually based on exponential forgetting.

2. Using the exponential forgetting factor explained above, we can formulate an efficient statistics maintenance method for our clustering method. Details of the maintenance method are described in Subsection 3.5.
3. By simply using one parameter λ to control the degree of weight decay in the document forgetting model, the information value of every document decays at the same rate. It provides a basis for the efficient implementation of our cluster maintenance method. Although we could use different λ values for different documents, such an approach would result in high processing cost, which is not suited to on-line environments.

3.3 Similarity measure based on document forgetting model

In this subsection, we derive the document similarity measure based on a probabilistic model by considering the document forgetting model introduced above.

In the following, we represent the documents in a document set by d_i $(i = 1, \ldots, n)$ and all the index terms in the document set by t_k $(k = 1, \ldots, m)$. We assume that the acquisition time of the documents $d_1, d_2, \ldots, d_n$ satisfies the relationship $T_1 \leq T_2 \leq \cdots \leq T_n$.

Definition 2 (Document selection probability) Let $\Pr(d_i)$ be the *subjective probability* to randomly select a document d_i from the document set. We define $\Pr(d_i)$ as follows:

$$\Pr(d_i) \stackrel{\text{def}}{=} \frac{dw_i}{tdw}, \tag{3}$$

where dw_i is the weight of d_i shown in Eq. 2 and tdw is the total weight of all documents in the document set:

$$tdw \stackrel{\text{def}}{=} \sum_{i=1}^{n} dw_i. \tag{4}$$

The selection probability of a document is proportional to its weight. This means that old documents have smaller selection probabilities than newer ones. The probability plays an important role in incorporating the notion of forgetting in the similarity function.

We next derive the conditional probability $\Pr(t_k|d_i)$ whose term t_k is selected from document d_i. We simply derive the probability based on the number of occurrences of terms in a document.

Definition 3 (Term occurrence probability) Let t_k be an index term, f_{ik} be the number of occurrences of term t_k within document d_i. Then the conditional probability $\Pr(t_k|d_i)$ that t_k is selected from d_i based on a random selection is

$$\Pr(t_k|d_i) \stackrel{\text{def}}{=} \frac{f_{ik}}{\sum_{l=1}^{m} f_{il}}. \tag{5}$$

The occurrence probability of t_k in the entire document set can be derived by

$$\Pr(t_k) = \sum_{i=1}^{n} \Pr(t_k|d_i) \cdot \Pr(d_i). \tag{6}$$

Using the above formulas and the Bayes' theorem, we obtain

$$\Pr(d_j|t_k) = \frac{\Pr(t_k|d_j)\Pr(d_j)}{\Pr(t_k)}. \tag{7}$$

Now we consider the conditional probability $\Pr(d_j|d_i)$. It can be expanded as

$$\Pr(d_j|d_i) = \sum_{k=1}^{m} \Pr(d_j|d_i,\ t_k)\Pr(t_k|d_i). \tag{8}$$

We make an assumption that $\Pr(d_j|d_i,\ t_k) \simeq \Pr(d_j|t_k)$ is approximately hold, then we get

$$\Pr(d_j|d_i) \simeq \sum_{k=1}^{m} \Pr(d_j|t_k)\Pr(t_k|d_i). \tag{9}$$

Based on the above formulas, we also get

$$\Pr(d_i,\ d_j) = \Pr(d_j|d_i) \cdot \Pr(d_i) \tag{10}$$

$$\simeq \frac{\Pr(d_i)\Pr(d_j)}{\sum_{l=1}^{m} f_{il} \sum_{l=1}^{m} f_{jl}} \sum_{k=1}^{m} \frac{f_{ik} f_{jk}}{\Pr(t_k)}. \tag{11}$$

This formula says that the co-occurrence probability between the two documents is based on their novelty, basically implied by $\Pr(d_i)$ and $\Pr(d_j)$, and the contents of the documents.

Next, we transform the above formulas to more simple ones using vector representation.

Definition 4 (Vector representation) The document vector $\boldsymbol{d}_i$ of d_i is defined as

$$\boldsymbol{d}_i \stackrel{\text{def}}{=} (tf_{i1} \cdot idf_1,\ tf_{i2} \cdot idf_2, \ldots, tf_{im} \cdot idf_m), \tag{12}$$

where tf_{ik} is the *term frequency* of t_k within d_i

$$tf_{ik} \stackrel{\text{def}}{=} f_{ik}, \tag{13}$$

and idf_k is the *inverse document frequency* (IDF) of t_k

$$idf_k \stackrel{\text{def}}{=} \frac{1}{\sqrt{\Pr(t_k)}}. \tag{14}$$

Let len_i be the *document length* of d_i:

$$len_i \stackrel{\text{def}}{=} \sum_{l=1}^{m} f_{il}. \tag{15}$$

Using the vector representation, Eq. 11 can be transformed as:

$$\Pr(d_i, d_j) = \Pr(d_i) \Pr(d_j) \frac{\boldsymbol{d}_i \cdot \boldsymbol{d}_j}{len_i \times len_j}. \tag{16}$$

This co-occurrence probability $\Pr(d_i, d_j)$ is derived based on the notion of document novelty and the $tf \cdot idf$ weighting scheme of the conventional vector space model.

Finally, we define the similarity metric as follows:

Definition 5 (Similarity measure) Given two documents d_i and d_j, their *similarity score* is defined as

$$sim(d_i, d_j) \stackrel{\text{def}}{=} \Pr(d_i, d_j). \tag{17}$$

This definition says the co-occurrence probability between d_i and d_j is used for their similarity score. $\Pr(d_i, d_j)$ is a probability to select d_i and d_j when we select two documents randomly from the document repository. The probability will be large when two documents have similar term occurrence patterns and they have recent timestamps. On the other hand, the probability will be small when two documents do not share same terms and/or at least one of the documents is old. That means when a document is old, the chance the document finds similar documents is quite rare. Therefore, such an old document tends to be an outlier when we perform clustering. As described later, our clustering algorithm carefully excludes the influences of outliers so that clustering focusing on recent documents can be attained.

In summary, our definition of document similarity—namely, document co-occurrence probability—assumes the following document-pair selection process:

1. When we select a pair of documents randomly from the document repository, two documents with similar contents (term occurrence patterns) have more chance to be selected.

This is a natural assumption in information retrieval. We have derived a variation of the $tf \cdot idf$ weighting of the conventional vector space model in a probabilistic manner. In our framework, however, we consider an additional assumption.

2. A random document selection probability (Eq. 3), which influences the document-pair selection probability, depends on the age of the document. The probability is determined by the forgetting model shown in Eq. 2.

Incorporation of the second assumption is an inherent feature of our approach. The assumption is incorporated as a *subjective probability* of document selection into the probabilistic similarity derivation. The idea of exponential forgetting is inherited from the idea in informetrics, and the appropriateness of the model is examined in the experimental evaluation.

3.4 Intuitive parameter setting

Our clustering method has two important parameters:

- *Forgetting factor* λ ($0 < \lambda < 1$): It specifies the speed of forgetting and is introduced in Subsection 3.2.
- *Expiration parameter* ε ($0 < \varepsilon < 1$): This parameter is used as a threshold value to control the clustering process. When the weight of a document (dw_i) becomes smaller than ε ($dw_i \leq \varepsilon$), the document is omitted from the clustering process.

To help decide these parameters, we use the following metaphors to impart intuitive meanings. To set the parameters λ and ε, we assume that the user gives a β value and a γ value, respectively. Their definitions are given below.

Definition 6 (Half-life span and life span) A *half-life span* parameter β specifies the period that a document loses half of its weight. Namely, β satisfies $\lambda^{\beta} = 1/2$. A *life span* parameter γ specifies the period that a document is "active" as the target of clustering.

Then, the forgetting factor λ and the expiration parameter ε can be derived as follows:

$$\lambda = \exp\left(-\frac{\log 2}{\beta}\right) \tag{18}$$

$$\varepsilon = \lambda^{\gamma}. \tag{19}$$

A *half-life span* is a traditional concept in citation analysis [13, 15]. It is a measure of how long articles in a journal continue to be cited after publication. In our approach, we borrowed the idea from citation analysis since it provides a more intuitive way for ordinary users. For example, the parameter setting $\beta = 7$ days is easier to interpret than its equivalent parameter setting $\lambda = 0.91$.

The notion of a *life span* is also an intuitive way to specify the expiration period. Since our clustering method clusters a time series of documents continually using novelty-based similarity, older documents become obsolete and do not contribute to the clustering result. Therefore, the expiration of obsolete documents is effective to reduce the processing cost and it does not degrade clustering quality.

The procedure to delete obsolete documents is covered in Appendix B.

3.5 Efficient updates of statistics and probabilities

We introduced the document similarity measure in Subsection 3.3. Since some of the statistics and probabilities used in its definition (e.g., $\Pr(d_i)$ and $\Pr(t_k)$) change their values when time has passed and when new documents are incorporated into

the document set, we have to recompute their new values. Since the recomputation becomes costly for large data sets, we devise an update method based on an incremental computation. It fully uses the previous statistics and probabilities values to achieve efficient updates. In practice, the statistics values at each update are stored persistently for use in subsequent updates.

Because the exponential forgetting factor λ is used, we can incrementally update the statistics and probabilities used in our clustering method. The formulas at the current time of update of these statistics and probabilities can be rewritten into the function of the forgetting factor λ or the function of the statistics and probabilities of the previous update. Since the values of the statistics and probabilities of the previous update are available, we can compute the update by reusing the previous statistics incrementally with low update overhead.

For example, assume that the last update was performed at $t = \tau$ and that new documents are incorporated at $t = \tau + \Delta\tau$. To compute a new clustering result at $t = \tau + \Delta\tau$, we need statistics values. Statistics for new documents should be obtained, but we need additional care since some statistics values (dw_i, tdw, $\Pr(d_i)$, and $\Pr(t_k)$, etc.) change over time. If we store previous statistics values, say $dw_i|_\tau$, we can calculate new statistics values incrementally such as:

$$dw_i|_{\tau+\Delta\tau} = \lambda^{\tau+\Delta\tau-T_i} = \lambda^{\Delta\tau} dw_i|_\tau$$

This is possible because of the exponential forgetting formula. Extending this idea, we can achieve low statistics update cost, which is linear in terms of the number of documents (n) and the number of index terms (m). In contrast, the naive approach, which computes an update from scratch, requires cost proportional to $n \times m$.

Formulation of incremental update is detailed in Appendix A.

4 Clustering algorithm

4.1 K-means method

The K-means method is a commonly used clustering method in information retrieval and other related research areas. By iteration, the method tries to refine the clusters of the previous iteration. The general algorithm is as follows:

1. Select K documents randomly as initial K clusters then generate initial cluster representatives.
2. Compare each remaining document with the cluster representatives and assign it to the most appropriate cluster.
3. When there is no change to the cluster assignment result, terminate the procedure. Otherwise, recompute the cluster representatives and return to Step 2.

The basic algorithm is quite simple; however, we need to clarify the following points clearly for practical implementation:

- The definition of cluster representatives,
- The criteria to select the most appropriate cluster in Step 2,
- The convergence condition of clustering used in Step 3.

In our clustering algorithm, we consider the extension to Steps 2 and 3 of the original K-means method.

4.2 Clustering index

The clustering index is a measure used to control the convergence criterion of clustering. At each iteration, the index is used to evaluate the quality of the clustering and to decide whether to stop the clustering process.

Definition 7 (Clustering index) Let G be the *clustering index*, K be the number of clusters to be generated and $|C_p|$ be the number of documents in cluster C_p. G is defined as:

$$G \stackrel{\text{def}}{=} \sum_{p=1}^{K} |C_p| \cdot avg\_sim(C_p), \tag{20}$$

where $avg\_sim(C_p)$ is the *average similarity* of documents in cluster C_p and is defined as:

$$avg\_sim(C_p) \stackrel{\text{def}}{=} \frac{1}{|C_p|(|C_p|-1)} \sum_{d_i \in C_p} \sum_{d_j \in C_p,\, d_i \neq d_j} sim(d_i,\, d_j). \tag{21}$$

The *intra-cluster similarity*, $avg\_sim(C_p)$, is used as a measure to decide the goodness and poorness of a clustering result.

4.3 Clustering procedure

The clustering algorithm is shown in Figure 2. It introduces a clear criterion for clustering convergence and for handling outliers. The handling of outliers is especially important in our clustering method because most old documents have low similarities with other documents due to the process of forgetting, and therefore tend to become outliers. The clustering method excludes documents that do not contribute to improved cluster quality. Documents put in the outlier list are regarded as normal documents in the next iteration because they may not fall in the outlier list again next time, since the contents of clusters will change.

In Step 1 of the iteration process of the clustering method, the computation overhead of the average similarity, *avg_sim*, shown in Eq. 21 is very large. We have developed an efficient calculation method for it using cluster representatives by extending the idea of Scatter/Gather [12]. Details are given in Appendix C.

5 Experimental evaluation

This section describes the experiments to evaluate performance of our approach with the TDT2 corpus. In this evaluation, we also considered comparing the performance of our clustering method with other conventional document clustering methods such as a hierarchical agglomerative clustering method (e.g., Scatter/Gather [12]) and a sequential clustering method (e.g., [36]). However, these methods are not suited for our clustering context. The hierarchical clustering method works very slowly. In our

- **Initial process**

  1. Select K documents randomly and form initial K clusters.
  2. Compute cluster representatives.
  3. Compute intra-cluster similarities and clustering index G.

- **Iteration process**

  1. For each document d, do the following two steps:
     (a) For each cluster, compute the intra-cluster similarity when d is appended to the cluster.
     (b) Assign d to a cluster such that the assignment causes the largest increase in the intra-cluster similarity. If no assignment increases the intra-cluster similarity, put d into the outlier list.
  2. Recompute cluster representatives.
  3. Recompute G and take it as G_{new} .
  4. If $(G_{\text{new}} - G_{\text{old}})/G_{\text{old}} < \delta$, terminate, where δ is a pre-defined constant.
  5. Otherwise, return to Step 1.

Figure 2 Clustering algorithm.

clustering context, new documents are appended and obsolete documents are deleted periodically. Reconstruction of the cluster hierarchy at each update, however, is very costly. The sequential clustering method, on the other hand, allows incremental incorporation of new documents. When new documents are appended, the method decides whether it should assign each document to an existing cluster or create a new cluster. However, this method does not consider deletion of documents. Thus it is difficult to present current "hot" clusters to users.

5.1 Dataset

The original TDT2 corpus [1] consists of chronologically ordered news articles obtained from six newswire sources and TV/radio broadcast services, ABC, APW, CNN, NYT, PRI and VOA, from January 4th to June 30th, 1998. The TDT2 corpus was developed by the Linguistic Data Consortium (LDC) [23]. It was used to evaluate the performance of approaches taking part in the topic detection and tracking (TDT) competition program in 1998 organized by the National Institute of Standards and Technology (NIST) [29].

There are 64,398 documents in the corpus and topics are assigned to documents to describe their topics. 96 topics were selected randomly. Beyond that there are two levels of relevance in a topic assignment: *fully relevant* ("YES") and *slightly relevant* ("BRIEF"). However, only 11,201 documents were labeled with the topics. In addition, we found that many documents among the annotated documents are

marked with more than one label. In this experiment, therefore, we use only those documents marked with one "YES" label. There are 7,578 documents corresponding to 96 topics dated from January 4th to June 30th, 1998 obtained by this selection. We call this TDT2 subset "selected TDT2 corpus." Topics in the selected TDT2 corpus are presented in Appendix D.

5.2 Evaluation of efficiency

The objective of this evaluation is to compare the efficiency of our incremental clustering with conventional non-incremental clustering.

5.2.1 Evaluation method

The selected TDT2 dataset is split into six contiguous and non-overlapping time windows. Each time window consists of news stories over 30 days, except for the last window which comprises only 28 days. The first to sixth time windows correspond to the period Jan4–Feb2, Feb3–Mar4, Mar5–Apr3, Apr4–May3, May4–Jun2, and Jun3–Jun30, respectively. Statistics of the divided time window are given in Table 2.

In this experiment, we consider two types of update processes:

Non-incremental process: All documents in each time window are used as an input.

Incremental process: The system incrementally updates the clustering result when new documents are delivered. To model this process, non-incremental clustering is first performed on the first Jan4–Feb2 time window as an initial step. After the initial clustering, the incremental process is used. The documents in the selected TDT2 corpus from February 3rd to June 30th are incrementally given with three days as an input unit.

For both processes, clustering is performed using two sets of parameters:

- Half-life span $\beta = 7$ days, life span $\gamma = 30$ days and $K = 24$,
- Half-life span $\beta = 30$ days, life span $\gamma = 30$ days and $K = 24$.

Table 2 Statistics for 30-day time window of selected TDT2 corpus.

	First	Second	Third	Fourth	Fifth	Sixth
No. of docs	1,820	2,393	823	570	1,090	882
No. of topics	30	44	47	39	40	43
Min. topic size	1	1	1	1	1	1
Max. topic size	461	875	129	96	327	138
Med. topic size	16.5	6	4	5	4.5	4
Mean topic size	60.67	54.39	17.51	14.62	27.25	20.51

These half-life span values, $\beta = 7$ days and $\beta = 30$ days, correspond to forgetting factor values $\lambda = 0.91$ and $\lambda = 0.98$, respectively. Choosing parameters with quite different values may provide clear insight into the effect of the half-life span on the performance of the clustering method. In addition, the life span $\gamma = 30$ days enables all documents to stay active during the clustering period since the 30-day time window length is used.

The experiment is run on a PC with a 3.2 GHz Pentium 4 CPU and 1 GB of RAM. The program is written using the Ruby programming language and executed in the Cygwin environment.

5.2.2 Evaluation results

To evaluate the efficiency of the incremental and non-incremental approaches, the computation times of statistics update and clustering consumed by the two processes are compared.

Tables 3 and 4 show the computation times required by the non-incremental and incremental clustering approaches for $\beta = 7$ days and $\beta = 30$ days, respectively. Computation time for the incremental process is the average computation time required by the incremental process to execute the three-day dataset in each time window. In the first column of the tables, "IP" stands for the incremental process and "NIP" is short for the non-incremental process.

The tables suggest that with the incremental process, we can achieve faster statistics update and clustering time in general. In statistics update, the computation time is approximately proportional to the number of documents to be updated. Since the number of documents to be updated by the incremental process is relatively small compared to the ones to be processed by the non-incremental process, the incremental process is more efficient than the non-incremental process. For clustering, the computation time depends heavily on the characteristics of the documents themselves and on the number of iterations. In incremental clustering, a new cluster structure does not change much from the previous structure even if a small number of documents are added and/or deleted, thus the incremental approach achieves faster computation time due to its fast convergence.

5.3 Evaluation of effectiveness

In this subsection, we evaluate effectiveness of the novelty-based clustering method. The objective of the evaluation is to compare effectiveness of the incremental

Table 3 Computation time in seconds ($\beta = 7$).

Dataset	Statistics update	Clustering
Feb3–Mar4 (IP/NIP)	135 / 1,585	581 / 939
Mar5–Apr3 (IP/NIP)	93 / 698	383 / 217
Apr4–May3 (IP/NIP)	48 / 535	89 / 220
May4–Jun2 (IP/NIP)	69 / 917	172 / 499
Jun3–Jun30 (IP/NIP)	63 / 712	180 / 337

Table 4 Computation time in seconds ($\beta = 30$).

Dataset	Statistics update	Clustering
Feb3–Mar4 (IP/NIP)	133 / 1,594	451 / 913
Mar5–Apr3 (IP/NIP)	89 / 674	265 / 239
Apr4–May3 (IP/NIP)	49 / 536	80 / 149
May4–Jun2 (IP/NIP)	72 / 887	134 / 256
Jun3–Jun30 (IP/NIP)	65 / 722	156 / 247

novelty-based clustering method with the non incremental method. Settings for the experiments are the same as those in Subsection 5.2.

5.3.1 Basic evaluation measures

We introduce the evaluation method for clustering performance used in the experiments. For each cluster, the documents in the cluster are compared with the selected TDT2 topics. The number of documents which correspond to each topic and are generated in the cluster is counted and represented by a. The remaining documents in the cluster which do not belong to the topic are counted and represented by b while c is the number of documents which discuss the topic but are not generated in the cluster. They are summarized in the following Table 5.

Clustering results are then evaluated by the following performance measures [4, 35]:

$$\text{Precision:} \quad p = \frac{a}{a+b} \tag{22}$$

$$\text{Recall:} \quad r = \frac{a}{a+c} \tag{23}$$

$$F_1 = \frac{2rp}{r+p} = \frac{2a}{2a+b+c} \tag{24}$$

F_1 is a harmonic mean of recall and precision. For each cluster, the precision, recall and F_1 are computed. We say that a cluster is *marked* with a topic if the precision of the topic in the cluster is equal to or greater than a predefined threshold. In the experiments, we use 0.60 as the threshold value. If a cluster does not have such a topic, the cluster is not marked with any topic.

We measure the global performance of our method by microaverage F_1 and macroaverage F_1 [35]. *Microaverage* F_1 is obtained by merging Table 5 for each marked cluster by summing the corresponding cells and then using the merged table to produce global performance scores. *Macroaverage* F_1 is obtained by producing

Table 5 Distribution of documents.

	On topic	Not on topic
In cluster	a	b
Not in cluster	c	d

per-cluster F_1 scores, then averaging the corresponding scores. These two measures are expressed by the following mathematical formulas:

$$\text{Microaverage } F_1 = \frac{\sum_{i=1}^{k} 2a_i}{\sum_{i=1}^{k} (2a_i + b_i + c_i)} \tag{25}$$

$$\text{Macroaverage } F_1 = \frac{1}{k} \sum_{i=1}^{k} F_1(c_i) \tag{26}$$

To assess the quality of clusters produced by the incremental and the non-incremental processes, the precision and recall and the macroaverage F_1 and microaverage F_1 are computed.

5.3.2 Evaluation results

Figures 3 and 4 show the macroaverage F_1 and microaverage F_1 scores of the incremental and non-incremental clustering results on each specific date using $\beta = 7$ days and $\beta = 30$ days, respectively. In the figure, the dates on the x-axis are the dates that the clustering results are observed.

These results show that the quality of clusters of the incremental process is generally better than the non-incremental process. The non-incremental approach consumes the 30-day dataset in an execution. The incremental approach, on the other hand, executes clustering on a 3-day basis. Thus, clustering effort of the incremental

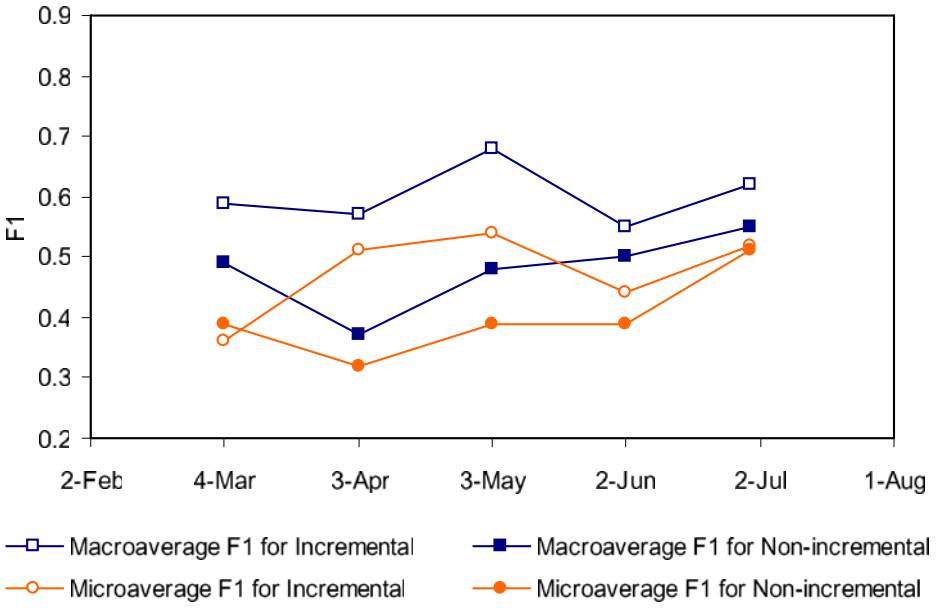

Figure 3 F_1 scores ($\beta = 7$).

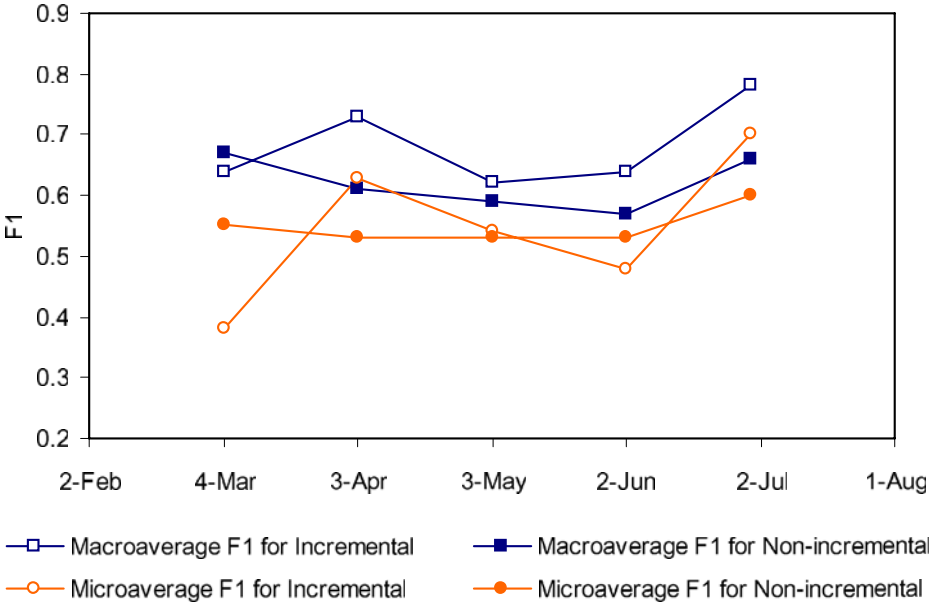

Figure 4 F_1 scores ($\beta = 30$).

approach is tenfold that of the non-incremental approach. With more executions, the incremental approach has chances to optimize the association between documents in the clusters and therefore produces better clustering results.

5.4 Evaluation of the effect of parameters

The objective of the evaluation is to examine the effect of parameters on the clustering method and to investigate appropriate K values for various values of the half life span parameter.

5.4.1 Evaluation criteria based on topic novelty

Since the goal of our clustering method is to generate clusters reflecting the trend of recent topics, recent topics must be identified. As the TDT2 evaluation dataset does not provide novelty information, this subsection introduces the evaluation method used in the experiments.

In the experiments, 60 days is considered a unit time window. A topic is judged as *recent* (R) if the topic has at least two documents in the interval 51st–60th day. Otherwise, if it has at least two documents in the interval 31st–50th day, it is judged as *less recent* (LR). If a topic is neither recent nor less recent, it is considered *old* (O). For example, according to the histogram in Figure 5, topic "Unabomber" is an old topic in the Jan4–Mar4 and May4–Jun30 time windows and a recent topic in the Mar5–May3 time window. In the histogram in Figure 6, topic "NBA finals" is a less recent topic in the May4–Jun30 time window.

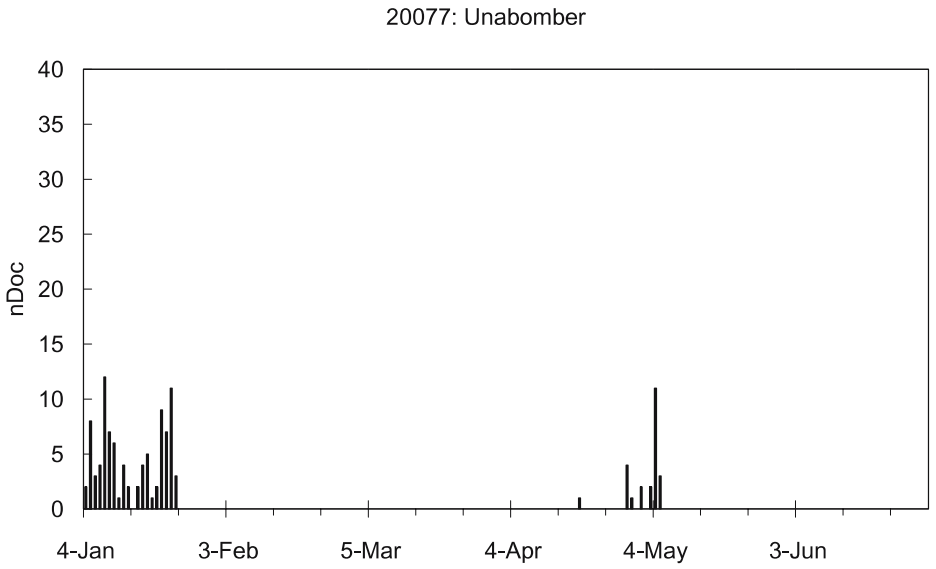

Figure 5 Histogram for topic 20077.

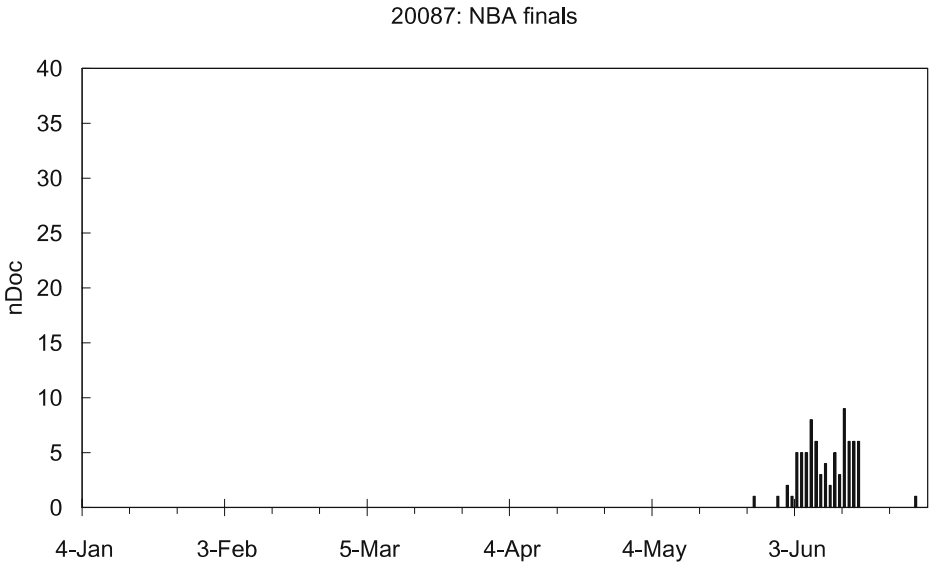

Figure 6 Histogram for topic 20087.

5.4.2 Evaluation method

The selected TDT2 corpus is divided into three contiguous and non-overlapping time windows, Jan4–Mar4, Mar5–May3, and May4–Jun30. Each time window consists of news stories of 60 days, except for the last time window, which has only 58 days. The statistics of the time windows are given in Table 6.

In this evaluation, we selected two half-life span values, $\beta = 7$ days and $\beta = 60$ days, which correspond to forgetting factor values $\lambda = 0.91$ and $\lambda = 0.99$, respectively. The idea behind this selection is that $\beta = 7$ assigns smaller weights to old documents and higher weights to recent ones in each time window, while $\beta = 60$ does not bias greatly toward recent topics. In other words, in 60 days, $\beta = 7$ causes weights of documents decayed from 0.91 to 0.003 and $\beta = 60$ from 0.99 to 0.5. The life span parameter is set to the same value, $\gamma = 60$ days, for each setting.

We use the cluster number settings $K = 8$, 12, 16 and 32 for $\beta = 7$ and $K = 16$, 24, 32 and 40 for $\beta = 60$ and then execute our method on the Jan4–Mar4 time window of the selected TDT2 dataset. The motivation behind the selection of smaller values of K for $\beta = 7$ and larger values of K for $\beta = 60$ is that $\beta = 7$ causes weights of old documents to be nearly zero and does the same for their similarity scores with other documents. That is, they become obsolete and inactive. It is almost impossible for these documents to join a cluster.

In the experiments, we use the non-incremental process of our method because the purpose of the experiments is to define appropriate parameter values for our clustering context. The experiments only require the final results when we have processed all documents in a time window. Therefore, the batch-oriented non-incremental version is suited to the experiments.

5.4.3 Evaluation results

Clustering results of the Jan4–Mar4 time window with different values of parameter K, which are evaluated by F_1 measures and novelty evaluation, are given in Table 7 for $\beta = 7$ and in Table 8 for $\beta = 60$.

As shown in Tables 7 and 8, we see that topics detected by $\beta = 7$ are mostly recent and some less recent, whereas $\beta = 60$ detects recent, less recent, and old topics.

For $\beta = 7$, topics generated by $K = 8$ and $K = 12$ are all recent topics and the number of a single topic detected in many different clusters is less than $K = 16$ and $K = 32$. In addition, the scores of macroaverage F_1 and microaverage F_1 are higher than $K = 16$ and $K = 32$. Therefore, we investigate the behavior of the clustering

Table 6 Statistics for 60-day time window of selected TDT2 corpus.

	Jan4–Mar4	Mar5–May3	May4–Jun30
No. of docs	4,213	1,393	1,972
No. of topics	51	61	54
Min. topic size	1	1	1
Max. topic size	1,251	189	414
Med. topic size	16	5	8
Mean topic size	82.61	22.84	36.52

Table 7 Clustering results ($\beta = 7$, Jan4–Mar4).

Topic ID	$K = 8$	$K = 12$	$K = 16$	$K = 32$	Recent?
20001	1	1	1	3	R
20002	1	1	2	2	R
20008				1	LR
20013	1	2	2	4	R
20015	1	1	3	6	R
20018		1	1	1	R
20020			1	1	R
20021	1	1	1	1	R
20022				1	LR
20023				1	R
20024				1	LR
20026	1	1	1	1	R
20032	1	1	1	1	R
20039	1	1	1	2	R
20040				1	R
20044			1	1	R
#of clusters	8	10	15	28	
Macro F_1	0.50	0.45	0.42	0.35	
Micro F_1	0.42	0.35	0.29	0.19	

method further by applying it on the other two time windows using $K = 8$ and $K = 12$ for $\beta = 7$. The results are given in Table 9 for the Mar5–May3 time window and Table 10 for the May4–Jun30 time window.

These results show that $K = 8$ produces higher F_1 values than $K = 12$, but $K = 12$ detects more recent topics than $K = 8$.

The clustering result for $\beta = 60$ in the Jan4–Mar4 time window (Table 8) shows that the macroaverage F_1 and microaverage F_1 of $K = 16$ are relatively high compared with other Ks. We think, however, that this K value is too small to enable enough topics to be generated. As mentioned in the preceding section, $\beta = 60$ causes the weights of documents decayed from 0.99 to 0.5 in 60 days. Namely, weights of documents by $\beta = 60$ do not decay very fast. In other words, they are still "highly active." Thus $K = 16$ is not considered further. On the other hand, for the parameter setting of $K = 40$, this K-value may be too large and causes same topics contained in different clusters. Hence, we examine the performance of $K = 24$ and $K = 32$ on the other two time windows, Mar5–May3 and May4–Jun30. The results are given in Tables 11 and 12.

Similar to the results of $\beta = 7$, these results show that $K = 24$ generates clusters of higher values of F_1 than $K = 32$, but $K = 32$ generally detects more recent topics than $K = 24$.

In summary, clustering results of $\beta = 7$ contain mostly recent topics and a few less recent ones, while $\beta = 60$ generates recent, less recent and old topics with higher F_1 scores compared to $\beta = 7$. Smaller values of K are more suitable for $\beta = 7$ and larger values are more appropriate for $\beta = 60$. In the experiments, $K = 8$ and $K = 12$ are suggested for $\beta = 7$ to obtain high quality and novelty results, and $K = 24$ and $K = 32$ are recommended for $\beta = 60$ to obtain high quality but less novelty results.

Table 8 Clustering results ($\beta = 60$, Jan4–Mar4).

Topic ID	$K = 16$	$K = 24$	$K = 32$	$K = 40$	Recent?
20001	1	4	5	5	R
20002	2	3	4	4	R
20004			1	1	O
20007				1	O
20008			1	1	LR
20009			1	1	LR
20012	1	1	1	1	LR
20013	2	2	3	3	R
20015	1	2	4	7	R
20018	1	1	1	1	R
20019	1	1	1	1	LR
20021			1	1	R
20022	1	1	1	1	LR
20023	1	1	1	1	R
20024			1	1	LR
20026	1	1	1	1	R
20031		1		1	LR
20032	1	1	1	1	R
20039	1	1	1	2	R
20044		1	1	1	R
20077	1	1		1	O
# of clusters	15	22	30	37	
Macro F_1	0.81	0.66	0.63	0.59	
Micro F_1	0.82	0.56	0.44	0.36	

5.5 Summary and discussion

In this evaluation, we have shown three types of evaluation methods to evaluate the performance of our clustering method; the first evaluation is aimed at comparing the efficiency of incremental and non-incremental clustering. The results from the experiments showed that incremental clustering is faster than non-incremental

Table 9 Clustering results ($\beta = 7$, Mar5–May3).

Topic ID	$K = 8$	$K = 12$	Recent?
20001		1	R
20002	1	1	R
20015	1	1	R
20044		1	R
20047	1	1	R
20065	1	1	R
20067		1	R
20071	1	1	LR
# of clusters	5	8	
Macro F_1	0.52	0.45	
Micro F_1	0.41	0.33	

Table 10 Clustering results ($\beta = 7$, May4–Jun30).

Topic ID	$K = 8$	$K = 12$	Recent?
20001	1	1	R
20002	1	2	R
20023	1	1	R
20044		1	R
20083	1	1	R
20086	1	1	R
20087	1	1	LR
20093		1	R
20096	1	1	R
# of clusters	7	10	
Macro F_1	0.71	0.64	
Micro F_1	0.64	0.57	

one; similar to the first evaluation, the second evaluation is intended to evaluate performance of the incremental and non-incremental clustering; however, here, it is in terms of the effectiveness of the two processes. The experimental results reveal that incremental clustering generally produces a better quality of clusters; in the third evaluation, several experiments were made to investigate the effect of parameters on the clustering method and to explore appropriate numbers of clusters, K, for different values of the half life span parameter. Results suggest that smaller values of

Table 11 Clustering results ($\beta = 60$, Mar5–May3).

Topic ID	$K = 24$	$K = 32$	Recent?
20001	3	2	R
20002	1	1	R
20015	1	3	R
20019	1	1	LR
20023	1	1	R
20024	1		O
20032	1	1	O
20039	1	1	O
20042	1	1	O
20044	2	3	R
20047	1	1	R
20048	1	2	LR
20056	1	1	R
20063		1	R
20065	2	2	R
20067		1	R
20071	2	2	R
20076		1	R
20077		1	R
# of clusters	20	26	
Macro F_1	0.72	0.68	
Micro F_1	0.70	0.63	

Table 12 Clustering results ($\beta = 60$, May4–Jun30).

Topic ID	$K = 24$	$K = 32$	Recent?
20001	3	3	R
20002	3	5	R
20004	1		LR
20005		1	O
20008	1		LR
20015		1	R
20019	1		LR
20023	1		R
20044	1	1	R
20047		1	R
20070	1	3	R
20071	1	2	LR
20072		1	LR
20074	1	1	LR
20076	1	1	LR
20077	1	1	O
20082		1	O
20086	2	2	R
20087	1	1	LR
20091	1	1	R
20093	1	1	R
20096	1	2	R
20098	1		O
# of clusters	23	29	
Macro F_1	0.77	0.66	
Micro F_1	0.74	0.57	

half life span generate mostly recent topics, while larger values produce recent, less recent, and old topics. Beyond that, to achieve high quality novelty-based clusters, the results also suggest that smaller values of K should be used for small values of half life span and larger values of K should be considered for larger values of half life span.

We observe that, in this evaluation, clustering results of $\beta = 60$ have higher F_1 scores than the results of $\beta = 7$. This occurs because $\beta = 7$ results in a steep decrease of document weights and makes it difficult for some documents to join a cluster. However, the F_1 measure does not consider "novelty" as is the case in our clustering context. Evaluating our method using F_1 measure results in low values of F_1 for small β clustering because many documents are forgotten in the clustering process, but are used for comparison with the documents in the evaluated dataset. We should regard F_1 scores, which do not consider novelty, as subordinary quality measures.

6 Conclusions and future work

In this paper, we have described our novelty-based document clustering method starting from the novelty-based similarity measure, the clustering algorithm, and

experimental evaluation. We have shown that the incremental algorithm of our approach exhibits good performance in the evaluation, both in terms of efficiency and effectiveness. Smaller half life span clustering performs better in detecting recent topics while the larger half life span one performs well in a general setting in which novelty of a topic is not considered. Thus the former is more suitable for our clustering context than the latter.

Further, we see that our method can answer our research problem; the method has shown that it is suited for an on-line setting where users are interested in acquiring new information. The method is more flexible and can be adapted to the user's requirement. Using a small forgetting factor, clustering results will contain mostly new documents. But if users are interested in obtaining clusters with better quality rather than new information, then they should use a large forgetting factor.

Our future work concerns the exploration of an evaluation measure that is better suited to our novelty-based clustering context. We also plan to investigate a method to show the entire process of the clustering method in a graphical user interface. This will make the system easier to use.

Acknowledgements This research is partly supported by the Grant-in-Aid for Scientific Research (16500048) from Japan Society for the Promotion of Science (JSPS), Japan, and the Grant-in-Aid for Scientific Research on Priority Areas (18049005) from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan. In addition, this work is supported by the grants from Kayamori Foundation of Information Science Advancement and Secom Science and Technology Foundation.

Appendix

A Update method of statistics and probabilities

Let the last update time of the given document set consisting of n documents $d_1, \ldots, d_n$ be $t = \tau$. Namely, the most recent documents are incorporated into the document set at $t = \tau$. Then suppose that n' new documents $d_{n+1}, \ldots, d_{n+n'}$ are appended at the time $t = \tau + \Delta\tau$. Therefore, their acquisition times are $T_{n+1} = \cdots = T_{n+n'} = \tau + \Delta\tau$. Let all the index terms contained in the document set at time $t = \tau$ be $t_1, \ldots, t_m$ and the additional terms incorporated by the insertion of documents $d_{n+1}, \ldots, d_{n+n'}$ be $t_{m+1}, \ldots, t_{m+m'}$. In the following discussion, we assume that $n \gg n'$ and $m \gg m'$ hold.

1. Updating of dw_is: First we consider the update of document weights of documents $d_1, \ldots, d_n$. We have already assigned a weight $dw_i|_\tau$ to each document d_i $(1 \leq i \leq n)$ at the last update time $t = \tau$. These weights have to be updated to $dw_i|_{\tau+\Delta\tau}$ in this update time. Since the relationship

$$dw_i|_{\tau+\Delta\tau} = \lambda^{\tau+\Delta\tau-T_i} = \lambda^{\Delta\tau} dw_i|_\tau \tag{27}$$

holds between $dw_i|_\tau$ and $dw_i|_{\tau+\Delta\tau}$, we can easily derive $dw_i|_{\tau+\Delta\tau}$ from $dw_i|_\tau$ by simply multiplying $\lambda^{\Delta\tau}$ to $dw_i|_\tau$. This property for the efficient update is due to the selection of the exponential forgetting factor in our document forgetting model.

For the new incoming documents $d_{n+1}, \ldots, d_{n+n'}$, we simply set $dw_i|_{\tau+\Delta\tau} = 1$ $(n+1 \le i \le n+n')$. The computational complexity of this step is estimated as $O(n+n') \approx O(n)$.

2. Updating of tdw: For the total weight of all the documents tdw, we can utilize the following update formula:

$$tdw|_{\tau+\Delta\tau} = \sum_{l=1}^{n+n'} \lambda^{\tau+\Delta\tau-T_l} = \lambda^{\Delta\tau} tdw|_\tau + n'. \tag{28}$$

The processing cost is $O(1)$.

3. Calculation of $\Pr(d_i)$s: $\Pr(d_i)$, the occurrence probability of document d_i, is given by

$$\Pr(d_i)|_{\tau+\Delta\tau} = \frac{dw_i|_{\tau+\Delta\tau}}{tdw|_{\tau+\Delta\tau}}. \tag{29}$$

Since we have already obtained $dw_i|_{\tau+\Delta\tau}$ and $tdw|_{\tau+\Delta\tau}$ in Steps 1 and 2, we can easily calculate $\Pr(d_i)$ when it is required.

4. Maintenance of $\Pr(t_k|d_i)$s: Since $\Pr(t_k|d_i)$ does not depend on time, we have to compute it only for the new documents $d_{n+1}, \ldots, d_{n+n'}$. If we roughly suppose that the number of terms contained in each document be a constant c, this step requires $O(cn') = O(n')$ computation time.

5. Updating of $\Pr(t_k)$s: The formula of $\Pr(t_k)|_\tau$ can be transformed as

$$\begin{aligned} \Pr(t_k)|_\tau &= \sum_{i=1}^{n} \frac{dw_i|_\tau}{tdw|_\tau} \cdot \Pr(t_k|d_i) \\ &= \frac{1}{tdw|_\tau} \sum_{i=1}^{n} dw_i|_\tau \cdot \Pr(t_k|d_i). \end{aligned} \tag{30}$$

Now we define $\widetilde{\Pr}(t_k)|_\tau$ as

$$\widetilde{\Pr}(t_k)|_\tau \stackrel{\text{def}}{=} \sum_{i=1}^{n} dw_i|_\tau \cdot \Pr(t_k|d_i), \tag{31}$$

then $\Pr(t_k)|_\tau$ is given by

$$\Pr(t_k)|_\tau = \frac{\widetilde{\Pr}(t_k)|_\tau}{tdw|_\tau}. \tag{32}$$

By storing $\widetilde{\Pr}(t_k)|_\tau$ instead of $\Pr(t_k)|_\tau$, we can achieve the incremental update. When we need the new value $\Pr(t_k)|_{\tau+\Delta\tau}$, we can compute it from $\widetilde{\Pr}(t_k)|_{\tau+\Delta\tau}$ and $tdw|_{\tau+\Delta\tau}$ using the above formula.

We can derive the update formula for $\widetilde{\Pr}(t_k)$:

$$\widetilde{\Pr}(t_k)|_{\tau+\Delta\tau} = \lambda^{\Delta\tau} \cdot \widetilde{\Pr}(t_k)|_\tau + \sum_{i=n+1}^{n+n'} \Pr(t_k|d_i). \tag{33}$$

Now we define $\Delta \Pr_{\text{sum}}(t_k)$ as

$$\Delta \underset{\text{sum}}{\Pr}(t_k) \stackrel{\text{def}}{=} \sum_{i=n+1}^{n+n'} \Pr(t_k|d_i), \tag{34}$$

then we get a simplified update formula

$$\widetilde{\Pr}(t_k)|_{\tau+\Delta\tau} = \lambda^{\Delta\tau} \cdot \widetilde{\Pr}(t_k)|_{\tau} + \Delta \underset{\text{sum}}{\Pr}(t_k). \tag{35}$$

Since it takes $O(n')$ time to compute a $\Delta \Pr_{\text{sum}}(t_k)$ value, we need $O(n' \cdot (m + m')) \approx O(n'm)$ time for all the documents.

Based on the above discussion, the total cost to update statistics and probabilities in an incremental manner is given by

$$O(n) + O(1) + O(n') + O(n'm) \approx O(n + n'm). \tag{36}$$

On the other hand, the naive scheme that calculate statistics and probabilities on each update has $O((n + n') \cdot (m + m')) \approx O(nm)$ computation time and is expensive for on-line document clustering applications.

Now we summarize the above ideas. We persistently store and incrementally maintain the following statistics: dw_is, tdw, and $\widetilde{\Pr}(t_k)$s, and achieve the update cost $O(n + m)$. Other statistics and probabilities ($\Pr(d_i)$s, $\Pr(t_k|d_i)$s, and $\Pr(t_k)$s) are computed when they are needed.

B Deletion of obsolete documents

To remove obsolete documents from the clustering target documents, we take the following approaches:

1. First we consider the deletion condition of old documents. In this paper, we take a simple approach: if the document weight dw_i for a document d_i satisfies the condition

   $$dw_i \leq \varepsilon \tag{37}$$

   for a small positive constant ε, we delete the document d_i. In practice, we delete the document weight dw_i, maintained as described in the previous section, from a persistent storage.
2. When we delete dw_i of the deleted document d_i, we have to propagate the deletion to other statistics. For tdw, the total weight of all the documents, we have to modify it as $tdw = tdw - dw_i$ according to its original definition. However, since now $dw_i \approx 0$, $tdw - dw_i \approx tdw$ holds so that we do not have to modify tdw actually.
3. We also need to delete f_{ik}s, the term occurrence frequencies for d_i, to reduce the storage cost. Therefore, we simply delete f_{ik}s for all the term t_ks that satisfy $f_{ik} > 0$.
4. Additionally, we have to delete $\widetilde{\Pr}(t_k)$ for each term t_k contained in d_i, but we should remind that the term t_k may be contained in other documents. In such a case, we should not delete these values because they are still active. To solve this problem, we simply use a reference counter for each term: when the reference counter becomes zero, we can safely delete the statistics values for the term.

C Update method of intra-cluster similarity

Let m be the total number of index term. The *cluster representative* $\boldsymbol{c}_p$ of cluster C_p is defined by

$$\boldsymbol{c}_p \stackrel{\text{def}}{=} (c_1^p,\ c_2^p, \ldots, c_m^p), \tag{38}$$

where

$$c_k^p \stackrel{\text{def}}{=} \sum_{d_i \in C_p} \frac{\Pr(d_i) \cdot tf_{ik} \cdot idf_k}{len_i} \quad (1 \le k \le m). \tag{39}$$

The *similarity* between cluster representatives of cluster C_p and C_q is defined by

$$cr\_sim(C_p,\ C_q) \stackrel{\text{def}}{=} \sum_{k=1}^{m} c_k^p c_k^q. \tag{40}$$

The *self similarity* of cluster representative of cluster C_p, $cr\_sim(C_p,\ C_p)$, can be expanded as

$$cr\_sim(C_p,\ C_p) = |C_p|(|C_p| - 1) \cdot avg\_sim(C_p) + ss(C_p), \tag{41}$$

where $ss(C_p)$ is the sum of the similarity of each document in cluster C_p with itself and defined as follows:

$$ss(C_p) \stackrel{\text{def}}{=} \sum_{d_i \in C_p} sim(d_i,\ d_i). \tag{42}$$

The average similarity in cluster C_p, $avg\_sim(C_p)$ shown in Eq. 21, can be written as

$$avg\_sim(C_p) = \frac{cr\_sim(C_p,\ C_p) - ss(C_p)}{|C_p|(|C_p| - 1)}. \tag{43}$$

If cluster $C_r = C_p \cup C_q$ and $C_p,\ C_q$ have no same elements ($C_p \cap C_q = \emptyset$), then

$$\begin{aligned} avg\_sim(C_r) = [&cr\_sim(C_p,\ C_p) + 2cr\_sim(C_p,\ C_q) \\ &+ cr\_sim(C_q,\ C_q) - ss(C_p) - ss(C_q)] \\ &/[(|C_p| + |C_q|)(|C_p| + |C_q| - 1)]. \end{aligned} \tag{44}$$

If C_q is a singleton cluster, that is $C_q = \{d_q\}$,

$$avg\_sim(C_r) = \frac{cr\_sim(C_p,\ C_p) + 2cr\_sim(C_p,\ C_q) - ss(C_p)}{|C_p|(|C_p| + 1)} \tag{45}$$

That is, to compute the avg_sim of an existing cluster C_p when d_q is appended to the cluster, we need to compute the similarity of cluster representatives $cr\_sim(C_p,\ C_q)$ only since $cr\_sim(C_p,\ C_p)$, $ss(C_p)$ and $|C_p|$ are computed once when cluster C_p is created and can be used as many times as required in one clustering iteration. By using the Eq. 45, we can reduce the cost to re-compute avg_sim when a document is appended to a cluster.

We can formulate similar update formulas for deletion when a document is removed from a cluster. They are omitted due to the space.

Table 13 Selected TDT2 corpus from Jan4–Jun30 1998.

Topic ID	Count	Topic name	Topic ID	Count	Topic name
20001	1034	Asian Economic Crisis	20026	70	Oprah Lawsuit
20002	923	Monica Lewinsky Case	20027	1	Pharoah's Tomb
20004	19	McVeigh's Navy Dismissal & Fight	20028	12	Mary Kay LeTourneau
20005	38	Upcoming Philippine Elections	20029	7	Buffett Buys Silver
20006	3	Israeli Palestinian Raids	20030	2	Pension for Mrs. Schindler
20007	15	Fossett's Balloon Ride	20031	36	John Glenn
20008	56	Casey Martin Sues PGA	20032	126	Sgt. Gene McKinney
20009	47	Karla Faye Tucker	20033	83	Superbowl '98
20010	7	Mountain Hikers Lost	20034	16	David Satcher Comfirmed
20011	18	State of the Union Address	20035	2	Holocaust Museum Resignation
20012	150	Pope visits Cuba	20036	5	Rev. Lyons Arrested
20013	530	1998 Winter Olympics	20037	33	Quality of Life, NYC
20014	2	African Leaders and World Bank Pres.	20038	1	LaSalle Boat FOUND
20015	1439	Current Conflict with Iraq	20039	119	India Parliamentary Elections
20016	6	$1 Million Stolen at WTC	20040	6	Tello (Maryland) Murder
20017	17	Babbitt Casino Case	20041	26	Grossberg baby murder
20018	99	Bombing AL Clinic	20042	29	Asteroid Coming??
20019	110	Cable Car Crash	20043	15	Dr. Spock Dies
20020	32	China Airlines Crash	20044	277	National Tobacco Settlement
20021	53	Tornado in Florida	20046	5	Great Lake Champlain??
20022	30	Diane Zamora	20047	93	Viagra Approval
20023	125	Violence in Algeria	20048	125	Jonesboro shooting
20024	38	Shevardnadze Assassination Attempt	20050	11	JJ the Whale
20025	1	Shoplifter's Hand Amputated	20052	6	Strike in Germany

Table 13 (continued)

Topic ID	Count	Topic name	Topic ID	Count	Topic name
20053	5	Capps Replacement Elections	20077	117	Unabomber
20054	1	Albright to Canada	20078	15	Denmark Strike
20055	1	Boeing Discrimination Suit	20079	8	Akin Birdal Shot & Wounded
20056	49	James Earl Ray's Retrial?	20080	1	Human Rights.Ethiopia
20057	7	World Figure Skating Champs	20081	1	Bad juice
20058	1	Guinness Gag	20082	4	Abortion clinic acid attacks
20059	1	UCONN Spring Weekend	20083	17	World AIDS Conference
20060	8	POW Memorial Museum	20084	5	Job incentives
20061	2	Kenya boosts Tourism	20085	8	Saudi Soccer coach sacked
20062	2	Mandela visits Angola	20086	138	GM Strike
20063	16	Bird Watchers Hostage	20087	79	NBA finals
20064	11	Race Relations Meetings	20088	5	Anti-Chinese Violence in Indonesia
20065	60	Rats in Space!	20089	23	Afghan Earthquake
20066	5	Marcus Allen Retires	20090	1	Unwed Fathers' Law
20067	7	Spanish Dam Broken	20091	51	German Train derails
20068	8	DiBella Treatment CURES Cancer?	20092	3	No Fat Drug
20069	3	Carter reunion	20093	12	Puerto Rico phone strike
20070	415	India, A Nuclear Power?	20094	5	Nazi-plundered Art
20071	201	Israeli-Palestinian Talks (London)	20095	4	Turkish Military Officers Fired
20072	1	Tony Awards	20096	64	Clinton-Jiang Debate
20073	1	Mother-Tongue Teaching	20097	2	Martin Fogel's law degree
20074	50	Nigerian Protest Violence	20098	9	Cubans returned home
20075	7	Food Stamps	20099	1	Oregon bomb for Clinton?
20076	225	Anti-Suharto Violence	20100	8	Goldman Sachs - going public?

D Selected TDT2 dataset

In this section, we show the complete selected TDT2 dataset in Table 13. In the table, we show Topic ID, Count which is the total number of documents contained in the topic, and Topic Name which corresponds to the Topic ID.

E Examples of the clustering results

In this section, we show some instances of the clustering results. Table 14 shows the result obtained on March 4, $K = 24$, $\beta = 7$ days, $\gamma = 30$ days, by using the incremental process mode. In Table 14, the *cluster#* is the system generated cluster number; # *of docs* is the total number of docs in each generated cluster; *Keywords* is the top ten high score keywords in each cluster; *TDT topic* is the corresponding TDT topic ID; *Recall* and *precision* is the values of the recall and precision measure of the cluster.

This example shows that our clustering approach exhibits good performance in terms of cluster quality as well. It generates high precision and relatively high recall except for some clusters which have low recall. In the above table, cluster# 4 is not

Table 14 Clustering results on March 4, $K = 24$, $\beta = 7$, $\gamma = 30$, IP.

Cluster#	# of docs	Keywords	TDT topic	Recall	Precision
0	14	georgian, shevardnadz, georgia, kidnapp, presid, attempt, observ, assassin, releas, suspect	20024	0.48	1.00
1	37	olymp, medal, snowboard, gold, marijuana, sport, rabagliatti, canadian, ross, test	20013	0.08	0.95
2	16	martin, tour, cart, pga, casei, golfer, profession, advantage, unfair, won	20008	0.70	1.00
3	196	olymp, medal, gold, won, skate, women, japan, nagano, game, winter	20013	0.44	1.00
4	20	protest, demonstr, abacha, student, ralli, govern, call, presid, dai, polic	(no topic)		
5	11	troop, gulf, iraq, georgia, persian, region, pentagon, ft, continu, anthrax	20015	0.01	1.00
6	143	iraq, agreem, weapon, council, annan, unit, secur, inspector, deal, secretari	20015	0.16	1.00
7	24	tucker, execut, death, texa, court, convict, suprem, karla, m, fay	20009	0.96	1.00
8	5	lyon, charge, church, nation, convent, baptist, theft, racket, investig, henri	20036	1.00	1.00
9	6	silver, price, buffett, investor, ounc, berkshir, market, compani, metal, percent	20029	1.00	1.00
10	9	kill, people, algeria, algier, bomb, algerian, attack, milit, islam, train	20023	0.39	1.00
11	46	iraq, kuwait, gulf, iraqi, militari, ship, oil, war, saddam, forc	20015	0.04	0.85

Table 14 (continued)

Cluster#	# of docs	Keywords	TDT topic	Recall	Precision
12	31	crash, airline, plane, taiwan, peopl, china, kill, airport, taipei, ground	20020	1.00	0.97
13	21	bomb, clinic, rudolph, atlanta, birmingham, alabama, eric, women, lyon, suspect	20018	0.36	1.00
14	25	winfrei, texa, beef, oprah, cattl, price, rancher, juri, cow, disea	20026	0.66	1.00
15	26	cabl, italian, investig, jet, marin, ski, babbitt, itali, fly, crew	20019	0.25	0.65
16	25	zamora, murder, sentenc, sheinbein, israel, kill, trial, fomer, prosecutor, life	20022	0.81	0.84
17	75	econom, indonesia, currenc, economi, presid, asian, asia, govern, crisis, suharto	20001	0.41	1.00
18	50	tornado, florida, peopl, central, victim, kill, week, damag, clinton, home	20021	0.96	1.00
19	58	hockei, team, olymp, czech, game, canada, player, gold, goal, republ	20013	0.13	1.00
20	58	lewinski, presid, clinton, jordan, monica, hous, white, investig, juri, grand	20002	0.22	1.00
21	520	iraq, presid, annan, iraqi, weapon, militari, secretari, unit, baghdad, saddam	20015	0.45	0.75
22	25	mckinnei, sexual, accus, sergeant, major, gene, armi, former, court, martial	20032	0.36	1.00
23	35	parti, india, elect, govern, seat, congress, bjp, vote, hindu, parliam	20039	0.54	1.00

marked with any TDT topic. Closer look into the result reveals that the cluster is a mixture of documents on many different topics. The precision of each topic in the cluster is not large enough to marked the topic to the cluster. In addition, some topics such as topic 20015 about "Current Conflict with Iraq" and 20013 about "1998 Winter Olympics" are marked to many clusters. This is because these topics are very large and "hot topics" at that time. Thus, the temporal weights and hence similarity scores between documents in the same topics become even stronger.

Similarly, we show another two clustering results in Tables 15 and 16. Tables 15 and 16 are the results for May4–Jun30 time window, $\beta = 7$, $\gamma = 60$, by using the non-incremental process with different K values; $K = 8$ for Table 15 and $K = 12$ for Table 16. Additional information, *norm_sim_val* value,[1] is added in the two

[1]Please note that the value of *norm_sim_val* should be smaller or equal to the value of number of documents in the cluster. However, in the results, the values of some clusters appear greater than the number of documents in the cluster. This is because, in the implementation, we omitted non-necessary computation which involves weighting a constant factor to all similarity scores while we are in a clustering process but does not contributes to the clustering results.

Table 15 Clustering results for $K = 8$, $\beta = 7$, $\gamma = 60$, May4–Jun30, NIP.

Cluster#	# of docs	Keywords	TDT topic	Recall	Precision	norm_sim_val
0	135	japan, asia, yen, economi, econom, market, financi, japanes, crisi, world	20001	0.54	0.80	34.99
1	89	bill, tobacco, lewinski, starr, presid, clinton, senat, hous, republican, monica	(no topic)			22.33
2	54	presid, clinton, china, chines, right, human, jiang, beij, tiananmen, squar	20096	0.85	0.98	68.67
3	92	strike, gm, worker, plant, motor, michigan, flint, north, unit, union	20086	0.60	0.90	69.04
4	32	lewinski, presid, tripp, grand, clinton, juri, hous, white, monica, former	20002	0.15	1	95.57
5	15	algeria, singer, milit, berber, matoub, muslim, report, kill, peopl, algier	20023	0.75	0.80	66.68
6	19	aid, drug, world, confer, viru, geneva, research, hiv, develop, report	20083	0.88	0.79	76.34
7	79	game, bull, chicago, saudi, jazz, jordan, team, utah, nba, world	20087	0.81	0.81	20.29

Table 16 Clustering results for $K = 12$, $\beta = 7$, $\gamma = 60$, May4–Jun30, NIP.

Cluster#	# of docs	Keywords	TDT topic	Recall	Precision	norm_sim_val
0	101	japan, asia, yen, economi, japanes, financi, crisi, market, econom, asian	20001	0.5	0.98	30.78
1	41	bill, tobacco, senat, republican, smoke, clinton, presid, democrat, compani, cigarett	20044	0.35	0.80	22.03
2	54	presid, clinton, china, chines, right, human, jiang, beij, tiananmen, squar	20096	0.85	0.98	68.67
3	81	strike, gm, plant, worker, motor, michigan, flint, north, unit, auto	20086	0.59	1	66.30
4	17	lewinski, tripp, presid, juri, grand, linda, clinton, monica, hous, white	20002	0.08	1	92.78

Table 16 (continued)

Cluster#	# of docs	Keywords	TDT topic	Recall	Precision	norm_sim_val
5	15	algeria, singer, milit, berber, matoub, muslim, report, kill, peopl, algier	20023	0.75	0.80	66.68
6	19	aid, drug, world, confer, viru, geneva, research, hiv, develop, report	20083	0.88	0.79	76.34
7	64	game, bull, chicago, jazz, jordan, utah, nba, citi, final, championship	20087	0.81	1	16.55
8	101	viagra, peopl, india, weapon, nuclear, drug, train, workder, iraq, report	(no topic)			12.40
9	16	saudi, citi, world, report, th, team, coach, rank, expens, viagra	(no topic)			28.10
10	6	telephon, puerto, rico, sale, dlr, compani, worker, sabotag, protest, million	20093	0.5	1	55.64
11	37	lewinski, presid, starr, clinton, hous, white, monica, lawyer, counsel, independ	20002	0.18	1	40.03

tables. This *norm_sim_val* is the intra-cluster similarity of a cluster. It is defined as a product of $|C_p| \cdot avg\_sim(C_p)$ where $|C_p|$ is the number of documents in cluster C_p and $avg\_sim(C_p)$ is the average similarity in cluster C_p. It appears in the formula of the clustering index G (Eq. 20) and represents how the cluster is appropriate in terms of self-similarity and cluster size.

References

1. Allan, J. (ed.): Topic Detection and Tracking: Event-based Information Organization. Kluwer, Boston, MA (2002)
2. Allan, J., Harding, S., Fisher, D., Bolivar, A., Guzman-Lara, S., Amstutz, P.: Taking topic detection and tracking from evaluation to practice. In: Proc. of the 38th Hawaii International Conference on System Sciences, pp. 1–10 (2005)
3. Avramescu, A.: Actuality and obsolescence of scientific literature. J. Am. Soc. Inf. Sci. **30**, 96–303 (1979)
4. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Harlow, England (1999)
5. Can, F.: Incremental clustering for dynamic information processing. ACM Trans. Inf. Sys. **11**(2), 143–164 (1993)
6. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: Proc. of 12th ACM SIGKDD Conference, pp. 554–560 (2006)
7. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental clustering and dynamic information retrieval. In: Proc. of 29th ACM Symposium on Theory of Computing (STOC), El Paso, Texas, USA, May 4–6, pp. 626–635 (1997)

8. Chaudhuri, B.B.: Dynamic clustering for time incremental data. Pattern Recogn. Lett. **15**(1), 27–34 (1994)
9. Chen, H.H., Kuo, J.J., Huang, S.J., Lin, C.J., Wung, H.C.: A summarization system for Chinese news from multiple sources. J. Am. Soc. Inf. Sci. Technol. (JASIST), **54**(13), 1224–1236 (2003)
10. Cohen, E., Strauss, M.: Maintaining time-decaying stream aggregates. In: Proc. of 20th ACM Symposium on Principles of Database Systems, San Diego, CA, June 9–11, pp. 223–233 (2003)
11. Cui, C., Kitagawa, H.: Topic activation analysis for document streams based on document arrival rate and relevance. In: Proc. of the 20th Annual ACM Symposium on Applied Computing, Santa Fe, NM, March 13–17, pp. 1089–1095 (2005)
12. Cutting, D., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/gather: A cluster-based approach to browsing large document collections. In: Proc. of 15th ACM SIGIR Conference, pp. 318–329 (1992)
13. Diodato, V.: Dictionary of Bibliometrics. Haworth Press, New York (1994)
14. Eichmann, D., Srinivasan, P.: Adaptive filtering of newswire stories using two-level clustering. Inf. Retr. **5**, 209–237 (2002)
15. Egghe, L., Rousseau, R.: Introduction to Informetrics: Quantitative Methods in Library, Documentation and Information Science. Elsevier, Amsterdam (1990)
16. Franz, M., McCarley, J.S., Ward, T., Zhu, W.J.: Unsupervised and supervised clustering for topic tracking. In: Proc. of ACM SIGIR Conference, pp. 310–317 (2001)
17. Ishikawa, Y., Chen, Y., Kitagawa, H.: An on-line document clustering method based on forgetting factors. In: Proc. of 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL), Darmstadt, Germany, September 4-9, pp. 325–339 (2001)
18. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**(3), 264–323 (1999)
19. Khy, S., Ishikawa, Y., Kitagawa, H.: Novelty-based incremental document clustering for on-line documents. In: Proc. of 2nd International Workshop on Challenges in Web Information Retrieval and Integration (WIRI), Atlanta, April 3, pp. 41–50 (2006)
20. Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proc. of ACM SIGKDD Conference, pp. 91–101 (2002)
21. Kumar, R., Novak, J., Raghavan, P., Tomkins, A.: On the bursty evolution of blogspace. World Wide Web J. **8**(2), 159–178 (2005)
22. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: Proc. of ACM SIGIR Conference, pp. 297–304 (2004)
23. Linguistic Data Consortium (LDC), http://www.ldc.upenn.edu/
24. Leuski, A., Allan, J.: Improving realism for topic tracking evaluation. In: Proc. of ACM SIGIR Conference, pp. 89–96 (2002)
25. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proc. of 5th Berkeley Symp. Math. Statist. Prob., vol. 1, pp. 281–297 (1967)
26. Mei, Q., Liu, C., Su, H.: A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In: Proc. of WWW Conference, pp. 533–542 (2006)
27. Mei, Q., Zhai, C.X.: Discovering evolutionary theme patterns from text—an exploration of temporal text mining. In: Proc. of SIGKDD Conference, pp. 198–207 (2005)
28. Nallapati, R., Feng, A., Peng, F., Allan, J.: Event threading within news topics. In: Proc. of 13th ACM CIKM Conference, pp. 446–453 (2004)
29. National Institute of Standards and Technology (NIST), http://www.nist.gov/speech/tests/tdt/
30. Radev, D., Otterbacher, J., Winkel, A., Blair-Goldensohn, S.: NewsInEssence, summarizing online news topics. In: Proc. of Communications of the ACM, pp. 95–98 (2005)
31. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
32. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading, MA (1989)
33. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv., **34**(1), 1–47 (2002)
34. Stokes, N., Carthy, J.: First story detection using a composite document representation. In: Proc. of the First International Conference on Human Language Technology Research, San Diego, CA, pp. 1–8 (2001)
35. van Rijsbergen, C.J.: Information Retrieval. Butter Worths, Sydney (1979)
36. Yang, Y., Pierce, T., Carbonell, J.G.: A study on retrospective and on-line event detection. In: Proc. of 21st ACM SIGIR Conference, pp. 28–36 (1998)

37. Yang, Y., Carbonell, J.G., Brown, R.G., Pierce, T., Archibald, B.T., Liu, X.: Learning approaches for detecting and tracking news event. IEEE Intel. Sys. Their Appl. **14**(4), 32–43 (1999)
38. Yang, Y., Zhang, J., Carbonell, J., Jin, C.: Topic-conditioned novelty detection. In: Proc. of ACM SIGKDD Conference, pp. 688–693 (2002)
39. Zhang, Y., Chu, C.H., Ji, X., Zha, H.: Correlating summarization of multi-source news with K-way graph bi-clustering. SIGKDD Explorations, **6**(2), 34–42 (2004)
40. Zhang, Y., Yu, J.X., Hou, J.: Web Communities Analysis and Construction. Springer, Berlin Heidelberg New York (2006)
41. Zhant, T., Ramakrishnan, R., Livny, M.: BIRCH: An efficient data clustering method for very large databases. In: Proc. of ACM SIGMOD Conference, pp. 103–114 (1996)