Developing a Document Clustering System Focusing on User Interest

Sophoin Khy[†1]

With the advent of the Internet technology, on-line documents such as news articles have become an important media for information dissemination. Since information becomes overwhelming when they are constantly delivered, it is difficult for a user to find useful information. Document clustering has been used as a core technique in managing vast amount of data and providing needed information. In addition, in on-line environments, users tend to be more 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 who is interested in recent issues. With this background, we proposed a novelty-based incremental clustering method for on-line documents that has biases on recent documents. In the clustering method, the notion of 'novelty' is incorporated into the similarity function and a clustering method, a variant of the K-means method, is proposed. In this report, the novelty-based incremental document clustering method is presented. The similarity measure, clustering algorithm, experimental evaluation of the clustering method and its implementation with the visualization system are described.

1. Introduction

With the advent of the Internet technology, electronic documents such as news articles, emails, weblogs, etc., have become an important media for information dissemination. This also led to the explosion of immense amount of on-line information on the Internet. On-line news articles which are delivered continually over the Internet is an example of such information explosion. With such profuse information, it is difficult for users to find useful information. Accordingly, intensive researches to mine important information have emerged[1),3),4),8)].

Document clustering is used as a core technique in managing vast amount of data. It is a method which groups documents into clusters such that documents in the same cluster are similar to each other, whereas documents in different clusters are dissimilar. It has been used as a fundamental method in many areas, such as information retrieval[6)], information filtering[7)], and topic detection and tracking[18)]. It has also been used as a preprocessing step for other document processing tasks, such as text classification[17)] and summarization of documents[16),20)].

In addition, in on-line environments, users are apt to be interested in new and up-to-date information. Although traditional document clustering methods can provide clusters of relevant documents to users to assist the browsing task, they do not fulfill the requirement of a user who is interested in recent issues.

With this background, we proposed a novelty-based incremental document clustering method which summarizes trends of on-line documents and provides users with up-to-date information[12),13)]. The objective of the method is to generate clusters reflecting trends of recent topics by presenting up-to-date cluster snapshots. It takes into consideration novelty of documents in the similarity measure and performs clustering based on an extension of the K-means method.

In this report, the novelty-based incremental document clustering method is presented. The similarity measure, clustering algorithm, experimental evaluation of the clustering method and its implementation with the visualization system are described.

The remainder of this report is organized as follows. Section 2 reviews related work. Section 3 describes the novelty-based incremental document clustering method. The system architecture and the visualization user interface of the method is presented in Section 4. Section 5 reports the experimental evaluation. Section 6 concludes the paper and discusses future work.

†1 Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba

2. Related Work

2.1 Topic Detection and Tracking

A research program relevant to our work is the *topic detection and tracking* (TDT)[22]. TDT tries to organize on-line documents, such as 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. Clustering approaches have been used in some TDT tasks. Research papers related to these tasks include[2),9),18)] for topic detection and topic tracking tasks, and[14),19)] 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.

2.2 Document Clustering

The *K-means* clustering method[11),15)], 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.

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^{2}ICM (cover-coefficient incremental clustering methodology)[5)], which is an incremental version of C^{3}M (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^{2}ICM enhances C^{3}M, 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[18)]. 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. The approach revealed relatively good performance in the TDT evaluation situation.

3. A Novelty-based Incremental Document Clustering Method

3.1 Similarity Measure

In this section, the novelty-based similarity function introduced in[10)] is described. It is derived from an aging model called the *document forgetting model*. The model is based on a simple intuition: the values of on-line documents delivered everyday are considered to be gradually losing their values as time passes.

The model introduces the notion of a *forgetting factor*. Every document is assigned an initial weight *one* when it is acquired from its source. The document weight gradually decays as time passes according to the rate specified by the forgetting factor. The *weight* of document d_i at time τ is defined as:

$$dw_i \equiv \lambda^{\tau - T_i}, \qquad (1)$$

in which λ ($0 < \lambda < 1$) is the *forgetting factor* and T_i ($T_i \leq \tau$) is the acquisition time of each document d_i. Figure 1 depicts the exponential decay of the document weight.

To set the parameter λ, we assume that the

user gives a *half-life span* value β. It specifies the period that a document loses half of its weight. Namely, β satisfies $\lambda^{\beta} = 1/2$. Therefore, λ can be derived as

$$\lambda = \exp(-\log 2/\beta). \tag{2}$$

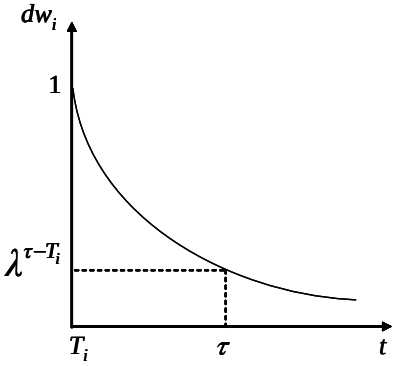

Fig. 1 Exponential decay function

If n is the number of documents in the repository, the *total weight* of all documents is:

$$tdw \equiv \sum_{l=1}^{n} dw_l. \tag{3}$$

We define the subjective probability that the document d_i is randomly selected from the document set as:

$$\Pr(d_i) \equiv \frac{dw_i}{tdw}. \tag{4}$$

That is, when a document is acquired from a news provider, the selection probability $\Pr(d_i)$ of the document is $1/tdw$. As time passes, the selection probability decreases and approaches zero. This selection probability indicates the effect that our approach is 'forgetting' old documents.

The conditional probability that a document d_j is obtained when d_i is given is:

$$\begin{aligned}\Pr(d_j|d_i) &= \sum_{k=1}^{n} \Pr(d_j|d_i,\, t_k)\Pr(t_k|d_i) \\ &\simeq \sum_{k=1}^{n} \Pr(d_j|t_k)\Pr(t_k|d_i),\end{aligned} \tag{5}$$

where t_k is an index term.

The co-occurrence probability of document d_i and d_j is:

$$\begin{aligned}\Pr(d_i,\, d_j) &= \Pr(d_j|d_i)\cdot\Pr(d_i) \\ &\simeq \Pr(d_i)\sum_{k=1}^{m}\Pr(d_j|t_k)\Pr(t_k|d_i).\end{aligned} \tag{6}$$

The $\Pr(t_k|d_i)$ used in above formula is the occurrence probability of term t_k in document d_i

$$\Pr(t_k|d_i) \equiv \frac{f_{ik}}{\sum_{l=1}^{m} f_{il}}, \tag{7}$$

where f_{ik} is the number of occurrences of term t_k in document d_i. $\Pr(d_j|t_k)$ can be defined by the Bayes' theorem as

$$\Pr(d_j|t_k) = \frac{\Pr(t_k|d_j)\Pr(d_j)}{\Pr(t_k)}. \tag{8}$$

If n is the total number of documents, the occurrence probability of term t_k, $\Pr(t_k)$, can be derived by

$$\Pr(t_k) \equiv \sum_{i=1}^{n}\Pr(t_k|d_i)\cdot\Pr(d_i). \tag{9}$$

Eq. (6) can be written as:

$$\Pr(d_i,\, d_j) \simeq \frac{\Pr(d_i)\Pr(d_j)}{\sum_{l=1}^{m} f_{il}\sum_{l=1}^{m} f_{jl}}\sum_{k=1}^{m}\frac{f_{ik}f_{jk}}{\Pr(t_k)}. \tag{10}$$

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.

If we represent document vector $\vec{d_i}$ of d_i by $tf\cdot idf$ weighting scheme

$$\vec{d_i} = (tf_{i1}\cdot idf_1,\, tf_{i2}\cdot idf_2, \ldots, tf_{im}\cdot idf_m) \tag{11}$$

and

$$tf_{ik} = f_{ik}, \tag{12}$$

$$idf_k = \frac{1}{\sqrt{\Pr(t_k)}}, \tag{13}$$

$$len_i = \sum_{l=1}^{m} f_{il}, \tag{14}$$

Eq. (6) can be transformed as:

$$\Pr(d_i,\, d_j) = \Pr(d_i)\Pr(d_j)\frac{\vec{d_i}\cdot\vec{d_j}}{len_i \times len_j}. \tag{15}$$

This co-occurrence probability of the two documents is defined as the *similarity score* between them

$$sim(d_i,\, d_j) \equiv \Pr(d_i,\, d_j). \tag{16}$$

From this similarity formula, the similarity

score between two documents is large when the two documents have similar term occurrence patterns and they have recent timestamps. It is small if the two documents do not share terms and/or at least one of the documents is old. By the incorporation of the forgetting factor in the similarity calculation, the clustering method clusters documents focusing on similarity and novelty of documents.

3.2 Clustering Algorithm based on *K*-means Method

3.2.1 *K*-means Clustering Method

The K-means method[11] is one of the commonly used clustering methods in data mining. 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.

3.2.2 Clustering Index

Our clustering algorithm introduces the *clustering index* G, which is computed by:

$$G \equiv \sum_{p=1}^{K} |C_p| \cdot avg\_sim(C_p), \tag{17}$$

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

$$avg\_sim(C_p) \equiv \frac{1}{|C_p|(|C_p|-1)} \sum_{d_i \in C_p} \sum_{d_j \in C_p,\, d_i \neq d_j} sim(d_i, d_j). \tag{18}$$

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

3.2.3 Proposed Clustering Algorithm

The proposed algorithm is an extension of the K-means algorithm. The K-means algorithm is extended considering the characteristics of the similarity formula shown above. A document is allocated to a cluster such that the assignment makes the largest increase in the *intra-cluster similarity*. The clustering algorithm is as follows.

- **Initial Process**
  (1) Select K documents randomly and form initial K clusters.
  (2) Compute cluster representatives.
  (3) Compute the intra-cluster similarities and the clustering index G.
- **Repetition 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 the cluster such that the increase of the intra-cluster similarity is the largest one. 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 $G_{\rm new}$.
  (4) If $(G_{\rm new} - G_{\rm old})/G_{\rm old} < \delta$, terminate, where δ is a pre-defined constant.
  (5) Otherwise, return to Step 1.

Documents put in the outlier list are regarded as normal documents in the next iteration since the documents may not fall in the outlier list next time as contents of clusters will change. The extended K-means method introduces a clearer criterion for clustering convergence and the handling of outliers, those documents which are considered not relevant to other documents in the clustering dataset.

In Step 1 of the repetition process, the computation overhead of the average similarity for each cluster, avg_sim shown in Eq. (18), each time a document is removed or appended to the cluster, is very large. The efficient computation of the avg_sim by using cluster representatives

is shown in[12),13)]. It is an extended idea of Scatter/Gather[6)].

3.3 Incremental Statistics Update and Clustering

In traditional document clustering, clustering is performed from scratch. However, since our target documents are on-line documents such as news articles which are delivered continually, we should take such dynamic nature into consideration.

The novelty-based document clustering method incorporates the incremental statistics update and incremental clustering processes.

3.3.1 Incremental Statistics Update

The values of some statistics and probabilities such as document weight dw_i, total weight of all documents tdw, the selection probability $Pr(d_i)$, etc., change with time and when new documents are incorporated into the document repository. Recalculating those statistics and probabilities from scratch tends to be costly for a large dataset. In this approach, the values of those statistics and probabilities are updated incrementally by using the values of previous statistics and probabilities to achieve efficient updates. The incremental calculation of some values will be introduced in this report. Details are described in[10),13)].

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

(1) Updating of dw_i's: First we consider the update of weights of documents $d_1, \ldots, d_m$. We have already assigned a weight $dw_i|_\tau$ to each document d_i ($1 \leq i \leq m$) 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

$$\begin{aligned} dw_i|_{\tau+\Delta\tau} &= \lambda^{\tau+\Delta\tau-T_i} \\ &= \lambda^{\Delta\tau} dw_i|_\tau \end{aligned} \tag{19}$$

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_{m+1}, ..., $d_{m+m'}$, we simply set $dw_i|_{\tau+\Delta\tau} = 1$ ($m+1 \leq i \leq m+m'$).

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

$$\begin{aligned} tdw|_{\tau+\Delta\tau} &= \sum_{l=1}^{m+m'} \lambda^{\tau+\Delta\tau-T_l} \\ &= \lambda^{\Delta\tau} tdw|_\tau + m'. \end{aligned} \tag{20}$$

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

$$\Pr(d_i)|_{\tau+\Delta\tau} = \frac{dw_i|_{\tau+\Delta\tau}}{tdw|_{\tau+\Delta\tau}}. \tag{21}$$

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

(4) Delete old documents d such that d satisfy $dw(d) < \epsilon$, where ϵ is a constant obtained from a user by specifying a *life span value* γ, the period that all documents in the document set are active. The value ϵ is derived as $\epsilon = \lambda^\gamma$. Then update statistics for all documents in the document set accordingly.

3.3.2 Incremental Clustering

The following incremental clustering procedure is proposed. Let τ be the timestamp when the previous clustering was performed, and $\tau' = \tau + \Delta\tau$ be the current timestamp.

(1) Incorporate new documents $d'_1, \ldots, d'_n$ arrived in the period $\tau \leq t \leq \tau'$ into the target document set.

(2) Perform clustering based on the proposed variant of the K-means clustering procedure shown above, but reuse the cluster representatives of the previous clustering performed at the timestamp τ and take them as the initial cluster representatives. The idea behind this is that the clustering tendency does not change greatly with minor modification to the target document set. Using the previous clustering results, the clustering process can be ac-

celerated.

4. System Architecture and Visualization

4.1 System Architecture

In this section, the architecture of the novelty-based incremental document clustering method is described. Figure 2 shows the system architecture of this approach.

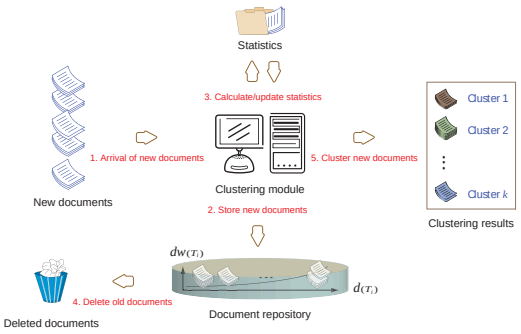

Fig. 2 System architecture

The clustering module consists of *statistics update* phase and *clustering* phase.

(1) *Statistics Update*: The statistics update phase is a pre-clustering processing phase. When new documents come, they are stored to the document repository. Probabilities and statistics necessary for the the computation of similarity scores are calculated before hand and are stored persistently to disk. This is because in this clustering approach, the similarity formula is based on the temporal notion; similarity scores change with time. The stored probabilities and statistics from previous computation are used to compute or update probabilities or statistics in the incremental process to achieve efficient computation cost. Obsolete documents are deleted and accordingly related probabilities and statistics are updated in this phase.

(2) *Clustering*: In this phase, similarity scores between documents are computed and the extended K-means clustering method are performed.

4.2 Visualization System

The visualization system for the novelty-based incremental document clustering method, called "Novelty-based Clusterer", provides a visualization interface of the clustering module to users, as shown in Figure 3. The visualization interface allows users to choose the parameter values. The snapshot of the visualization interface for parameter selection is shown in Figure 4. When the user selects a parameter set, clustering is performed based on the parameter set. The clustering process and the result at each time point is added as depicted in Figure 5. When the user selects to view a clustering result, the clustering snapshot is presented (Figure 6). In the clustering snapshot, the user can read the content of a news article by selecting the name of the news article s/he wants to read (Figure 7).

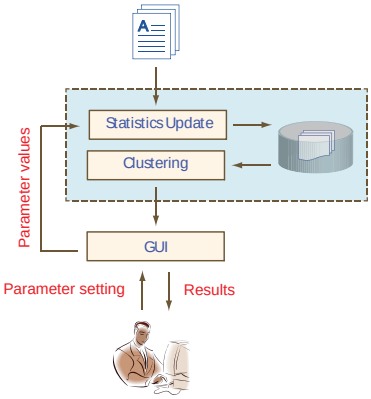

Fig. 3 The visualization interface and clustering module

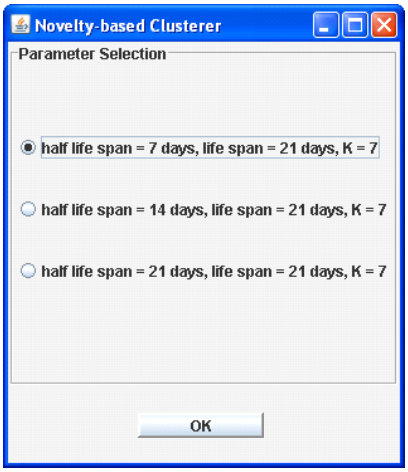

Fig. 4 Parameter selection

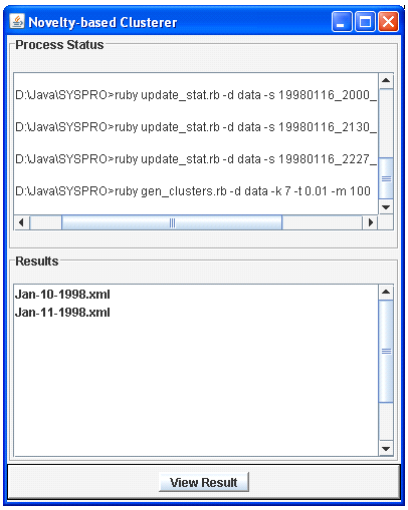

Fig. 5 Process and results

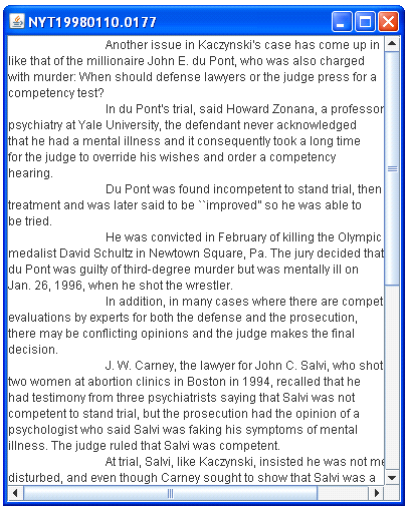

Fig. 7 A news article

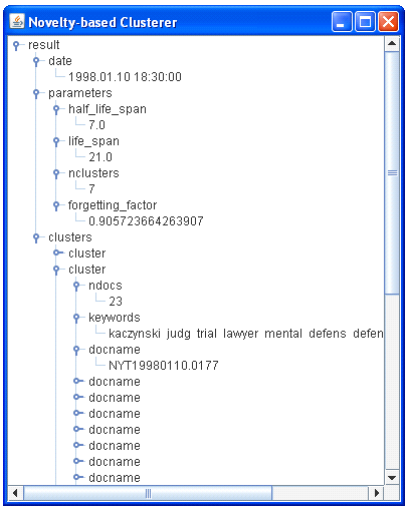

Fig. 6 A clustering result

5. Experimental Evaluation

To evaluate the performance of this method, various experiments were performed. However, due to space constraints, the experiments on effect of parameters forgetting factor and number of clusters K are not presented in this report. They are reported in reference[13].

In this report, the efficiency and the effectiveness of the incremental process and the non-incremental process are explored. Computation time and cluster quality of the two processes are assessed.

5.1 Dataset

The TDT2 corpus developed by the Linguistic Data Consortium[21] consists of chronologically ordered news articles obtained from newswire sources and TV/radio broadcast services. The corpus consists of 64,398 documents from January 4th to June 30th 1998. However, there are only 11,201 documents labeled with topics (96 topics) as "YES" and/or "BRIEF". In addition, we found that some documents among the annotated documents are marked with more than one label. Therefore, we selected only those documents marked with only one "YES" label and used in our experiments. There are 7,578 documents corresponding to 96 topics dated from January 4th to June 30th 1998 obtained by this selection. These TDT2 subset is called "selected TDT2 corpus". Some topics in the selected TDT2 corpus are presented in Table 1.

The selected TDT2 dataset is split into six contiguous and non-overlapping time windows. Each time window consists of data of 30 days, except for the last time window which consists of 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. The statistics of the divided time window is shown in Table 2.

5.2 Evaluation Measure

Clustering results are evaluated by the following performance measures[18]:

- Precision: $p = a/(a+b)$
- Recall: $r = a/(a+c)$
- $F_1 = 2rp/(r+p) = 2a/(2a+b+c)$

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

	First	Second	Third	Fourth	Fifth	Sixth
No. of docs	1820	2393	823	570	1090	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

Table 1 Some topics in selected TDT2 corpus from Jan4-Jun30 1998

Topic ID	Count	Topic Name
20001	1034	Asian Economic Crisis
20002	923	Monica Lewinsky Case
20004	19	McVeigh's Navy Dismissal & Fight
20011	18	State of the Union Address
20012	150	Pope visits Cuba
20013	530	1998 Winter Olympics
20015	1439	Current Conflict with Iraq
20018	99	Bombing AL Clinic
20026	70	Oprah Lawsuit
20033	83	Superbowl '98
20044	277	National Tobacco Settlement
20076	225	Anti-Suharto Violence
20077	117	Unabomber
20078	15	Denmark Strike
20082	4	Abortion clinic acid attacks
20083	17	World AIDS Conference
20086	138	GM Strike
20087	79	NBA finals
20088	5	Anti-Chinese Violence in Indonesia
20096	64	Clinton-Jiang Debate
20099	1	Oregon bomb for Clinton?
20100	8	Goldman Sachs - going public?

where a, b and c refer to the number of documents in each category in Table 3 respectively.

Table 3 Distribution of documents

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

For each clustering result, the system generated clusters are compared with the selected TDT2 topics and the precision and recall for each cluster are computed. Based on several observations, we define a cluster is marked with a topic if the precision of the topic in the cluster is equal to or greater than 0.60. If a cluster has no precision larger than 0.60, then the cluster is not marked with any topic.

Then we measure the global performance of our method by microaverage F_1 and macroaverage F_1[18]. F_1 is a harmonic mean of recall and precision. *Microaverage* F_1 is obtained by merging the Table 3 for each marked cluster by summing the corresponding cells and then using the merged table to produce global performance measures. *Macroaverage* F_1 is obtained by producing per-cluster performance measures, then averaging the corresponding measures[18]. These two measures are expressed by the following mathematical formulas:

$$\text{Macroaverage } F_1 = \frac{1}{k}\sum_{i=1}^{k} F_1(c_i) \qquad (22)$$

$$\text{Microaverage } F_1 = \frac{\sum_{i=1}^{k} 2a_i}{\sum_{i=1}^{k}(2a_i + b_i + c_i)} \qquad (23)$$

5.3 Experimental Setting

In this experiment, the number of clusters K = 24, life span γ = 30 days, and two half life span values, β = 7 days and 30 days, are selected. These half life span values 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 30-day life span will enable all documents to stay active during the clustering period since the 30-day time window is used.

Moreover, the same forgetting factor value is applied to all documents in a series of the clustering operations. That is all news articles are assumed to have the same aging speed regardless of which topics the articles are about. Choosing different forgetting factor values is not feasible in this clustering approach since we do not know in advance which topics a document belongs to before we group them and get clustering results. Moreover, use of the same forgetting factor enables the achievement of efficiency in incremental processing of the clustering approach.

5.4 Experimental Framework

The experimental framework is designed as follows.

(1) Non-incremental process: The six time window dataset described in the previous section is used as an input to this process.

(2) Incremental process: This approach is commonly adopted in the real world setting in which a system has already accumulated data and generated clusters. Then the incremental process is used to incrementally update the clustering results when new documents are continually delivered to the system. By modeling this procedure, the non-incremental clustering is performed on the first, Jan4-Feb2, time window data described in the previous section as a preliminary step for this experiment. After the preliminary clustering, the incremental process is adopted. The data in the selected TDT2 corpus from February 3rd to June 30th are incrementally and continually given as three-day input data to the clustering system using the incremental process which consists of incremental statistics update process and incremental clustering process.

(3) For both processes, clustering is performed using two sets of parameters:
- half life span = 7 days, life span = 30 days and $K = 24$,
- half life span = 30 days, life span = 30 days and $K = 24$.

The reason behind the selection of the three-day input data for the incremental process is that the number of documents in the selected TDT2 corpus used in the experiment is small, 7,578 documents, and spans over a long period of time from January 4th to June 30th, 1998. Hence the number of documents contained in one day is very small. Therefore, three-day data is used.

5.5 Experimental Results and Discussion

The experiment is performed on a PC with Pentium 4 CPU, speed 3.2 GHz and 1 GB of RAM. The program is written using Ruby programming language and implemented on Cygwin.

5.5.1 Evaluation of Efficiency

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

Table 4 and Table 5 show the computation time in seconds required by the non-incremental and incremental processes for 7-day half life span and 30-day half life span respectively. The computation time for 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 the *NIP* is short for the non-incremental one.

The non-incremental process takes thirty-day dataset, one time window, as its input, whereas the incremental process takes as its input three-day dataset a time. The evaluation of the efficiency of both processes here is not to compare the computation time of the process on thirty-day data with the process on three-day data. The idea is that rather than performing clustering from scratch every time new documents are received, adopting the incremental process, efficient execution time can be achieved.

The tables suggest that using the incremental process, in general, we can achieve faster statistics update and clustering time. In statistics update, the computation time is approximately proportional to the number of the documents to be updated. Since the number of documents to be updated by the incremental process is relatively small compared to the number of documents to be processed by the non-incremental process, the incremental process is more efficient than the non-incremental one. For clustering, the computation time depends heavily on the characteristics of the documents themselves and on the number of iterations. For incremental clustering, the new cluster structure is thought to not change much from the previous structure even if small number of documents are added and thus it achieves faster computation time.

5.5.2 Evaluation of Effectiveness

To assess the quality of clusters produced by the incremental and the non-incremental processes, clustering results are compared with the selected TDT2 topics and the precision and recall and the macroaverage F_1 and microaverage F_1 are computed.

Figure 8 and Figure 9 show the macroaverage F_1 and microaverage F_1 scores of the incremental and the non-incremental clustering results at each specific date using 7-day half life span and 30-day half life span respectively. In the

Table 4 Computation time of 7-day half life span (in seconds)

Dataset	Statistics Updating	Clustering
Feb3-Mar4 (IP/NIP)	135 / 1585	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 5 Computation time of 30-day half life span (in seconds)

Dataset	Statistics Updating	Clustering
Feb3-Mar4 (IP/NIP)	133 / 1594	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

figures, the dates on the x-axis are the dates that the clustering results are observed.

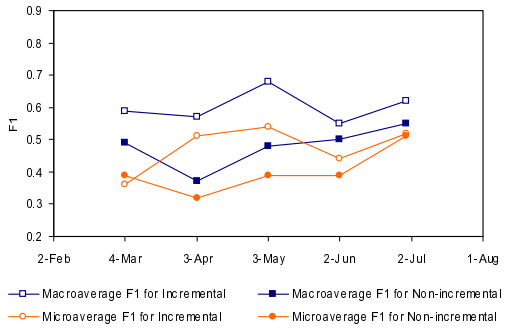

Fig. 8 F1 scores for 7-day half life span

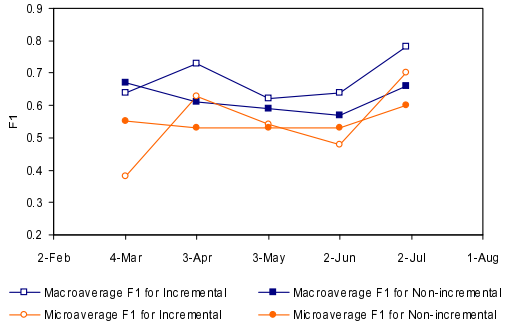

Fig. 9 F1 scores for 30-day half life span

These results show that the quality of clusters of the incremental process is generally better than the non-incremental process. The non-incremental approach takes thirty-day dataset, one time window, as its input and processes them once. However, the incremental process takes as its input three-day dataset a time. Thus it takes ten times for the incremental approach to process data as much as the non-incremental one. Each clustering of the ten times probably gradually optimizes the association between documents in the clusters and results in better clustering results in the incremental process.

6. Conclusions and Future Work

In this report, the novelty-based similarity measure, clustering algorithm, system architecture and visualization, and experimental evaluation of the novelty-based incremental document clustering method are described. The experiments have shown that the incremental algorithm of this approach exhibits good performance, in terms of both efficiency and effectiveness.

For future work, the exploration of an evaluation measure that is better suited to the novelty-based clustering context than the recall, precision and F_1 measures is very useful in the evaluation of aging function based clustering methods. In addition, the exploration of a framework to summarize the overall trend in the large collection of accumulated document clustering results is an interesting research direction.

References

1) Allan, J. (ed.): Topic Detection and Tracking: Event-based Information Organization. Kluwer, Boston (2002)
2) Allan, J., Harding, S., Fisher, D., Bolivar, A., Guzman-Lara, S., Amstutz, P.: Taking Topic Detection and Tracking from Evaluation to Practice. Proc. of the 38th Hawaii Interna-

tional Conference on System Sciences, pp. 1–10 (2005)
3) Baeza-Yates, R., and Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Harlow, England (1999)
4) Berry, W. M. (ed): Survey of Text Mining: Clustering, Classification, and Retrieval. Springer (2003)
5) Can, F.: Incremental Clustering for Dynamic Information Processing. ACM Trans. Inf. Sys. **11**(2), pp. 143–164 (1993)
6) Cutting, D., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. Proc. of 15th ACM SIGIR conference, pp. 318–329 (1992)
7) Eichmann, D., Srinivasan, P.: Adaptive Filtering of Newswire Stories using Two-Level Clustering. Information Retrieval, **5**, pp. 209–237 (2002)
8) Feldman, R., Sanger, J.: The Text Mining Handbook. Cambridge University Press, Sao Paulo (2007)
9) Franz, M., McCarley, J.S., Ward, T., Zhu, W.J.: Unsupervised and Supervised Clustering for Topic Tracking. Proc. of ACM SIGIR Conference, pp. 310–317 (2001)
10) Ishikawa, Y., Chen, Y., Kitagawa, H.: An Online Document Clustering Method Based on Forgetting Factors. Proc. of 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL), Darmstadt, Germany, September 4-9, pp. 325–339, (2001)
11) Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys **31**(3), (1999)
12) Khy, S., Ishikawa, Y., Kitagawa, H.: Novelty-based Incremental Document Clustering for On-line Documents. Proc. of 2nd International Workshop on Challenges in Web Information Retrieval and Integration (WIRI), Atlanta, April 3, pp. 41–50 (2006)
13) Khy, S., Ishikawa, Y., Kitagawa, H.: A Novelty-based Clustering Method for On-line Documents, World Wide Web Journal, DOI 10.1007/s11280-007-0018-9 (2007)
14) Kumaran, G., Allan, J.: Text Classification and Named Entities for New Event Detection. Proc. of ACM SIGIR Conference, pp. 297–304 (2004)
15) MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. Proc. of 5th Berkeley Symp. Math. Statist. Prob., **1**, pp. 281–297 (1967)
16) Radev, D., Otterbacher, J., Winkel, A., Blair-Goldensohn, S.: NewsInEssence, Summarizing Online News Topics. Proc. of Communications of the ACM, pp. 95–98 (2005)
17) Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, **34**(1), pp. 1–47 (2002)
18) 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. **14**(4), July/August, pp. 32–43 (1999)
19) Yang, Y., Zhang, J., Carbonell, J., Jin, C.: Topic-conditioned Novelty Detection. Proc. of ACM SIGKDD Conference, pp. 688–693 (2002)
20) 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), pp. 34–42 (2004)
21) Linguistic Data Consortium (LDC), http://www.ldc.upenn.edu/
22) Topic Detection and Tracking (TDT), http://www.nist.gov/speech/tests/tdt/