Incremental Clustering Based on Novelty of On-line Documents

Sophoin KHY♡ Yoshiharu ISHIKAWA♢
Hiroyuki KITAGAWA♠

Clustering has been widely used as a fundamental method in many areas such as characterization and classification. Various clustering researches have been conducted since decades ago. In previous papers, we presented a novelty-based incremental document clustering method which considers novelty of on-line documents in its similarity measure and performs clustering based on an extended algorithm of the *K*-means method. This paper further examines the performance of the incremental and non-incremental processing of the clustering method and effect of parameter values on the method by showing the experimental results.

1. Introduction

The proliferation of on-line information services that distribute news, emails, weblogs, etc., has led to the increase of vast amount of on-line information sources on the Internet and the requirement for managing and extracting useful information from myriad electronic documents is on the rise. Moreover, in on-line environments, users are apt to be interested in new and up-to-date information. However, traditional clustering methods groups similar documents into clusters by assigning same temporal importance to all documents.

Our research focuses on a clustering method which has biases on recent documents, namely, novelty-based document clustering. In previous papers, we proposed a novelty-based document clustering method which considers novelty of on-line documents in its similarity function and performs clustering based on an extended K-means method [4, 5].

This paper examines the behavior of our clustering method by conducting two experiments. The first experiment is intended to evaluate the efficiency and effectiveness of the incremental and the non-incremental processes. The second experiment is for investigating the effect of parameters on the clustering method.

♡ Student Member. Graduate School of System and Information Engineering, University of Tsukuba. sophoin@kde.cs.tsukuba.ac.jp

♢ Regular Member. Information Technology Center, Nagoya University. ishikawa@itc.nagoya-u.ac.jp

♠ Regular Member. Graduate School of System and Information Engineering / Center for Computational Science, University of Tsukuba. kitagawa@cs.tsukuba.ac.jp

2. Related Work

Topic Detection and Tracking (TDT) [1] is a research project organized by NIST [3]. TDT evaluation competitions were held to evaluate research progress and technical capabilities. In some TDT research tasks defined in the TDT program [3] such as topic tracking, topic detection and new event detection, various clustering techniques are used to generate clusters of stories that discuss the same topic. On the other hand, the goal of our novelty-based clustering method is to generate clusters presenting an overview of current trend of recent topics. Since the goal of our research is different from the TDT research, we cannot use the TDT evaluation framework directly in our research.

3. Similarity Measure

In this section, we briefly provide an overview of the forgetting-factor-based similarity function introduced in F^{2}ICM [4] which is used in our clustering method. The similarity measure is derived from 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 *weight* of document d_i at time τ is defined as:

$$dw_i \equiv \lambda^{\tau - T_i}, \tag{1}$$

in which λ ($0 < \lambda < 1$) is the *forgetting factor* and T_i ($T_i \leq \tau$) is the acquisition time of document d_i. Each document is assigned an initial weight *one* at its acquisition time from it source.

To set the parameter λ, a user can give a *half-life span* value β specifying the period that a document loses half of its weight. Namely, β satisfies $\lambda^\beta = 1/2$. Then, λ can be derived as

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

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

$$\Pr(d_i) \equiv \frac{dw_i}{tdw}, \tag{3}$$

in which tdw is the *total weight* of all documents

$$tdw \equiv \sum_{l=1}^{n} dw_l. \tag{4}$$

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

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

Please see our previous paper [5] which introduces additional definitions of other probabilities.

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

$$sim(d_i, d_j) \equiv \Pr(d_i, d_j). \quad (6)$$

This definition is based on a probabilistic model that the similarity scores between document d_i and d_j is the process of searching for the two documents d_i and d_j in a document set such that the two documents are similar to each other. We calculate their co-occurrence probability as their similarity score.

From the similarity formula, we can say that the more a document d_i becomes old, the smaller its similarity scores with other documents as $\Pr(d_i)$ values of old documents are small.

4. Clustering Algorithm based on *K*-means Method

4.1 Clustering Index

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

$$G \equiv \sum_{p=1}^{K} |C_p| \cdot avg\_sim(C_p), \quad (7)$$

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

$$avg\_sim(C_p) \equiv \frac{1}{|C_p|(|C_p|-1)} \sum_{d_i \in C_p} \sum_{d_j \in C_p, d_i \neq d_j} sim(d_i, d_j). \quad (8)$$

4.2 Clustering Algorithm

The clustering algorithm used in this paper is an extension of the *K*-means algorithm. A document is allocated to a cluster such that the assignment makes the largest increase of *intra-cluster similarity*.

1. Select K documents randomly and form initial K clusters.
2. Compute cluster representatives.
3. Compute intra-cluster similarities and clustering index G.
4. 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 increase of the intra-cluster similarity is the largest. If no assignment increases the intra-cluster similarity, put d into an outlier list.
5. Recompute the cluster representatives.
6. Recompute G and take it as G_{new} .
7. If $(G_{\text{new}} - G_{\text{old}})/G_{\text{old}} < \delta$, terminate, where δ is a predefined constant. Otherwise, return to Step 4.

Documents put in the outlier list are regarded as normal documents in the next iteration since they may not be outliers next time since the contents of clusters change.

The extended *K*-means method introduces a clearer criterion for clustering convergence and handling of outliers. Our method also incorporates the incremental statistics updating process and incremetal clustering process for efficient update [4, 5].

5. Experiments

In this section, we describe our experimental methodology and results on TDT2 corpus.

5.1 Dataset

In TDT2 corpus [2], not all documents are labeled with topics. On the other hand, labeled documents belong to more than one topic. Thus, we selected only those documents marked with only one "YES" label and used in experiments. There are 7,578 documents of 96 topics dated from January 4th to June 30th 1998. These TDT2 subset is called "selected TDT2 corpus".

5.2 Evaluation Measure

The system generated clusters are compared with the selected TDT2 topics. Precision and recall [6] 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. In addition, we measure the global performance of our method by microaverage F_1 and macroaverage F_1 as introduced in [6]. F_1 is a harmonic mean of recall and precision.

5.3 Experiment 1

Our clustering method consists of two phases; statistics update and clustering. Statistics update computes and stores statistics and probabilities for clustering. This phase is necessary in our clustering method since our similarity function changes with time. So we need to store the statistics and probabilities for next computation. In this experiment, computation time and cluster quality of incremental and non-incremental processes for statistics update and clustering will be assessed.

Experimental Framework

The experimental framework is designed as follows.

- The selected TDT2 dataset is split into six time windows; Jan4-Feb2, Feb3-Mar4, Mar5-Apr3, Apr4-May3, May4-Jun2 and Jun3-Jun30.
- Non-incremental process: The six time window dataset is used as an input to this process.
- Incremental process: The non-incremental clus-

tering is performed on the first Jan4-Feb2 time window as a preliminary step for this experiment. Then the incremental process is adopted. The documents in the selected TDT2 corpus from February 3rd to June 30th are incrementally and continually given using three days as one unit. We use three-day data since the number of documents in the selected TDT2 in one day is too few.

- For both processes, clustering is performed using this parameter set: half life span β = 7 days, life span γ = 30 days and K = 24

The parameter life span γ is used to define the value of the parameter ϵ in $\epsilon = \lambda^{\gamma}$. If $dw_i < \epsilon$, the document will be deleted from the document repository. These parameter values are defined by users.

Experimental Results

The experiment is performed on a PC with Pentium 4 CPU 3.2 GHz and RAM 1 GB on Cygwin environment. The program is written in Ruby language.

Table 1 shows the computation time in seconds required by the non-incremental (*non-inc*) and incremental (*inc*) processes to run the statistics update and clustering for the 7-day half life span clustering. The computation time for incremental process is the average processing time to process a three-day unit in a time window.

Table 1: Computation time of 7-day half life span (sec)

Dataset	Stat. Update	Clustering
Feb3-Mar4 (inc / non-inc)	135 / 1585	581 / 939
Mar5-Apr3 (inc / non-inc)	93 / 698	383 / 217
Apr4-May3 (inc / non-inc)	48 / 535	89 / 220
May4-Jun2 (inc / non-inc)	69 / 917	172 / 499
Jun3-Jun30 (inc / non-inc)	63 / 712	180 / 337

The evaluation of the efficiency of both processes here is not to compare the computation time of the process on thirty-day data (non-inc) with the process on three-day data (inc). The idea is that rather than performing clustering from scratch every time new documents are recieved, adopting the incremental process, in general, efficient execution time for statistics update and clustering can be achieved.

Figure 1 shows the macroaverage F_1 and microaverage F_1 scores of the incremental and the non-incremental clustering at each specific date using the 7-day half life span clustering. The result shows the quality of clusters of the incremental process is generally better than the non-incremental process. 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 gradually optimizes the association between documents in the clusters and results in better clustering results for the incremental process.

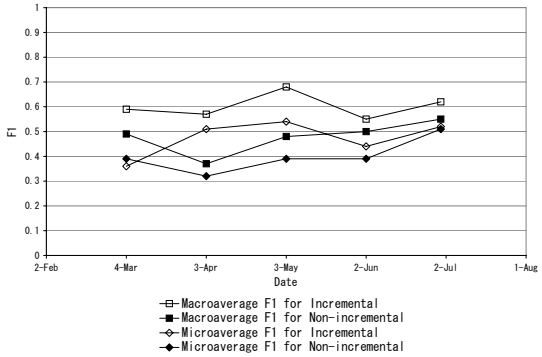

Figure 1: F1 scores for 7-day half life span

5.4 Experiment 2

The objective of this experiment is to examine the effect of half life span papameter on the clustering method and to investigate appropriate K values for different values of half life span.

The selected TDT2 corpus is divided into three time windows, Jan4-Mar4, Mar5-May3 and May4-Jun30. Since the goal of our clustering method is to generate clusters reflecting current trend of recent topics, it is necessary to identify what recent topics are. We introduce an evaluation concept which is used in this experiment. A topic is judged as *recent* (R) if the topic has at least two documents in the interval 51st-60th day. Or, if it has at least two documents in the interval 31st-50th day, it is judged as *less recent* (LR). Otherwise, it is regarded *old* (O). For example, by the definition, topic "Unabomber" (Figure 2) is a *recent* topic in Mar5-May3 and an *old* topic in Jan4-Mar4 and May4-Jun30.

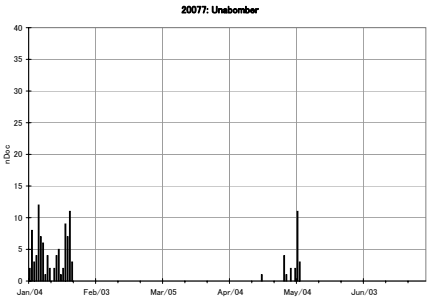

Figure 2: Histogram for topic 20077

Experimental Results

In this experiment, we selected two half life span values β, 7 days and 60 days corresponding to $\lambda = 0.91$ and $\lambda = 0.99$ respectively. We choose various K values, K = 8, 12, 16 and 32 for 7-day half life span, and K = 16, 24, 32 and 40 for 60-day half life span. In the experiments, we use the non-incremental process of our method as the purpose of the experiments is to define appropriate parameter values for our clustering con-

text. The experiments only need final results when we have processed all the documents in a time window. Therefore, batch-oriented non-incremental version is suited for the experiments.

Table 2 and Table 3 show the results for 7-day half life span and for 60-day half life span on the Jan4-Mar4 time window.

Table 2: Clustering results (β = 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 F1	0.50	0.45	0.42	0.35	
Micro F1	0.42	0.35	0.29	0.19	

Table 3: Clustering results (β = 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 F1	0.81	0.66	0.63	0.59	
Micro F1	0.82	0.56	0.44	0.36	

As shown in Table 2 and Table 3, we see that the 7-day half life span clustering detects mostly recent topics whereas the 60-day one detects recent, less recent and also old topics. Moreover, smaller K produces less number of topics but has higher F_1 values. Furthermore, the 60-day half life span clustering generates higher F_1 scores than the 7-day half life span one. This is because F_1 measure does not consider 'novelty' as in our clustering context.

6. Conclusions and Future Work

In this paper, we have introduced our novelty-based clustering method and results of experiments. We have shown that the incremental process is more efficient and effective than the non-incremental one. In addition, smaller half life span clustering performs better in detecting recent topics while larger one performs well in general setting in which novelty of topics is not considered.

Future work includes a method to automatically estimate the K values for different values of half life span parameters and investigation of an evaluation measure that is more suitable for our novelty-based clustering context than the recall, precision and F_1 measures.

[Acknowledgements]

This research is partly supported by the Grant-in-Aid for Scientific Research (15300027 and 16500048) from Japan Society for the Promotion of Science (JSPS), Japan, and the Grant-in-Aid for Scientific Research on Priority Areas (16016205) from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan. In addition, this work is supported by the grants from the Asahi Glass Foundation and the Inamori Foundation.

[References]

[1] J. Allan (ed.): *Topic Detection and Tracking: Event-based Information Organization*, Kluwer, 2002.

[2] http://www.ldc.upenn.edu/

[3] http://www.nist.gov/speech/tests/tdt/

[4] Y. Ishikawa, Y. Chen, and H. Kitagawa: "An On-line Document Clustering Method Based on Forgetting Factors", *Proc. of 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'01)*, pp. 325–339, 2001.

[5] Y. Ishikawa, and H. Kitagawa: "An Improved Approach to the Clustering Method Based on Forgetting Factors", *DBSJ Letters*, Vol. 2, No. 3, pp. 53–56, December 2003.

[6] Y. Yang, J.G. Carbonell, R.G. Brown, T. Pierce, B.T. Archibald, and X. Liu: "Learning Approaches for Detecting and Tracking News Event", *IEEE Intelligent Systems*, Vol. 14, No. 4, 1999.

Sophoin KHY
Sophoin Khy is a student of Graduate School of Systems and Information Engineering, University of Tsukuba. She received the M.Eng. degree from Master's Program in Sciences and Engineering, University of Tsukuba, in 2006. Her research interests include information retrieval, data and web mining, and databases. She is a student member of ACM SIGMOD Japan and DBSJ.

Yoshiharu ISHIKAWA
Yoshiharu Ishikawa is a Professor at Information Technology Center, Nagoya University. His research interests include databases, data and web mining, digital libraries and information retrieval. He is a member of DBSJ, IPSJ, IEICE, JSAI, ACM and IEEE Computer Society.

Hiroyuki KITAGAWA
Hiroyuki Kitagawa is a Professor at Graduate School of Systems and Information Engineering, University of Tsukuba. His research interests include integration of heterogeneous information sources, WWW and databases, structured documents, semi-structured data, multimedia databases, and human interface. He is a member of DBSJ, IPSJ, IEICE, JSSST, ACM and IEEE Computer Society.