I.J. Intelligent Systems and Applications, **2011, 3, 26-32**
Published Online May 2011 in MECS (http://www.mecs-press.org/)

New Trending Events Detection based on the Multi-Representation Index Tree Clustering

Hui Song
School of computer science and technology, Donghua University, Shanghai, China
Email: songhui@dhu.edu.cn

Lifeng Wang, Baiyan Li and Xiaoqiang Liu
School of computer science and technology, Donghua University, Shanghai, China
Email: { lfwang, libaiyan, liuxq}@ dhu.edu.cn

***Abstract*—Traditional Clustering is a powerful technique for revealing the hot topics among Web information. However, it failed to discover the trending events coming out gradually. In this paper, we propose a novel method to address this problem which is modeled as detecting the new cluster from time-streaming documents. Our approach concludes three parts: the cluster definition based on Multi-Representation Index Tree (MI-Tree), the new cluster detecting process and the metrics for measuring a new cluster. Compared with the traditional method, we process the newly coming data first and merge the old clustering tree into the new one. Our algorithm can avoid that the documents owning high similarity were assigned to different clusters. We designed and implemented a system for practical application, the experimental results on a variety of domains demonstrate that our algorithm can recognize new valuable cluster during the iteration process, and produce quality clusters.**

***Index Terms*—new trending events, incremental Clustering, Incremental priority, multi-representation index tree**

I. Introduction

With the explosion of online publishing in the World Wide Web, people have to browse a large data collection to locate valuable information. Many researches on intelligence Web Mining have been developed, e.g. "Trending" or "Hot" topic detection is helpful to learn what the focus of attention is in Social Networks. Text clustering technique had been widely involved into this research which classifies the documents containing related contents into one cluster, and ranking the number of documents in each cluster can give the hot topics [1]. To deal with online dataset, incremental text clustering has been used [2].

In practical application the challenge is: usualness events with large documents always dominate the ranking list, such as popular singers, matches. For example, in public security department, most reports talk about stealing, robbery, sharper and, etc. but are there any new type events? Is there something new needed to be focused on? We call this intention "new trending events". They are hard to be detected with incremental clustering algorithm because the total number of such data is small and these data always come gradually, they may be inserted into different clusters during the input process.

Detecting the new trending events can be modeled as such a problem: detecting the new clusters from the document stream. It resembles to the task of New Event Detection (NED) in Topic Detection and Tracking (TDT) program, but differs in: NED is defined as detecting the first story of a topic in time-streaming news [3], but the new cluster detection tends to find a set of documents reporting new event, not just one new story.

In this paper, we present a novel process to address this problem. We proposed a model based on increment priority clustering method, including: new event definition, new cluster detecting process and new cluster measurement. The new cluster detection process composed with three steps: (1) we cluster new documents into a Multi-Representation Indexing Tree (MI-Tree). (2) Merge the old index tree into the new one. (3) Recognize the new clusters after the merging step. We have built an integrated text analysis platform, and this algorithm is embedded into the platform to test the efficiency. Experiments on the different data sets show that our contributions in this work as follows:

(1) We formally model the document cluster of an event as a Multi-Representation Indexing Tree, which is a concise statistical Multi-representation of indexing tree. The hierarchical indexing tree diagrammatically reveals the events and the sub-events relationship, and we represent a cluster with multi-points to deal with non-spherical data.

(2) We give preferential treatment to incremental data, which is different with classical incremental clustering method. It avoids that the document of new cluster are scattered into old clusters, and can recognize new type of events in time with stable accuracy.

The rest of this paper is organized as follows. Section 2 gives a review of related work in incremental clustering and NED. In section 3, we introduce our model of the new event detection. We explain the detail detection process in section 4 and Section 5. Section 6 describes the experimental results, and conclusions with future work are wrapped up in Section 7.

II. Related Works

Most event detection works address tasks defined in Topic Detection and Tracking (TDT), including new event detection, topic tracking and retrospective event detection [3, 4]. The goal is to group documents (e.g., news articles) received from one or more temporally-ordered stream(s) according to the events that they describe. Our work is based on new event detection and topic tracking research, and tends to detect the new trending event which has gathered some documents and gains increasing attention gradually.

The common approach is modeling event detection as an online incremental clustering task [5, 6]. Documents reached from a stream are processed in the order of their arrival. For each document, its similarities to the existing events (clusters of documents) are computed, and the document is assigned to either an existing event or a new event based on predefined criteria. Methods in this approach vary mainly in the way of computing the similarity between a document and an existing event [7, 8].

There are many published algorithms which aimed to incrementally cluster points in a data set, including DC-tree clustering [9], incremental hierarchical clustering [10], et al.

Khaled M. Hammouda [11] proposed SHC incremental clustering, which relies only on pair-wise document similarity information. Clusters are represented with a Cluster Similarity Histogram. A concise statistical representation of the distribution of similarities within each cluster provides a measure of cohesiveness. But the time complexity of SHC is $O(n^2)$, since it must compute the similarity to all previously seen documents for each new one. Chung-Chian Hsu [12] proposed M-ART and the conceptual hierarchy tree to solve similar degrees of mixed data.

As [2] pointed out, by identifying broad and narrow clusters and describing the relationship between them hierarchical clustering algorithms generate knowledge of topic and subtopic, so incremental clustering based hierarchical method is widely researched. Maria Soledad [13] clustered the RSS news articles, gave available similarity metrics of RSS articles. Ref. [14] implemented a novel hierarchical algorithm called LAIR2, which has constant running time average for on-the-fly Scatter/Gather browsing.

Zhang Kuo[5] represented an incremental clustering based on news indexing-tree created dynamically. Indexing-tree is created by assembling similar stories together to form news clusters in different hierarchies according to their values of similarity. Comparisons between current document and previous clusters could help finding the most similar document in less comparing times. It performs the clustering process efficiently with $O(n)$ time complexity. But it prefers spherical data and leads to lower accuracy caused by class center decentralization. Our work contributes on improving the accuracy of indexing-tree without increasing computation time.

The approach of our work is more related to [15], it propose a novel automatic online algorithm for news issue construction, through which news issues can be automatically constructed with real-time update, and lots of human efforts will be released from tedious manual work. However, it didn't consider complicated hierarchies situation. In many cases, it is hard to detect the sub-cluster's change. Our work gives detail approach based on MI-TREE hierarchical algorithm.

III. New Cluster Detection Model

For a dynamic increasing document dataset, we describe the clustering result formally as an index tree for the existing text dataset. While the new documents is coming in a certain period of time, we cluster them as a new index tree, then merge the previous tree into it and re-insert earlier outliers into it. In this process, the clusters of the whole dataset have also been updated.

In our model, the bag-of-word approach is adopted to record the features of a document; the clustering result of a dataset is described as a multi-representation index tree, the similarity and new cluster metric are given based on these definitions.

A. Pre-Processing and Page Representation

The primary step of text clustering is the extraction of features from documents to create a term vector for each document, followed by clustering or grouping based on those features.

Incremental TF-IDF model is widely applied in term weight calculation. TF-IWF model [16] is chosen to weight terms for its steadier performance in many experiments. WF (word frequency) of term w at time t is calculated as:

$$wf_t(w) = wf_{t-1}(w) + wf_{St}(w) \qquad (1)$$

where S_t means a set of documents coming at time t, and $wf_{St}(w)$ means the appearance number of term w appears in the newly appearing documents. $Wf_{t-1}(w)$ represents the appearance number of term w appears before time t. As showed in formula (1), WF is updated dynamically at time t.

Each document d coming at time t is represented as an n-dimension vector, where n is the number of distinct terms in document d. Each dimension is weighted using incremental TF-IWF model and the vector is normalized so that it is of unit length:

$$weight_t(d,w) = \frac{tf(d,w)\log((W_t+1)/(wf_t(w)+0.5))}{\sqrt{\sum_{w'\in d}(tf(d,w')\log((W_t+1)/(wf_t(w')+0.5)))^2}} \qquad (2)$$

where $tf(d,w)$ means how many times term w appears in document d and W_t represents the total appearance number of the term before time t:

$$W_t = \sum_{t_d \le t}\sum_{w'\in d} tf(d,w') \qquad (3)$$

t_d in formula (3) means that document d appears at time t.

B. Multi-reprentation Index Tree

For various types of clustering method, they exploit their own cluster representation. The classical hierarchical algorithm CURE applied heap structure and K-d tree to represent the generated cluster and the representative point of a cluster. Here we define a multi-representation index tree to achieve the same goal.

The index tree is defined formally as follows:

$$MI\text{-}Tree = \{r, N_C, N_d, N_a, E\}$$

where r is the root of *MI-Tree*, N_C is the set of all cluster nodes, N_d is the set of document nodes which have been assigned to a cluster, N_a is the set of document nodes isolated to any clusters, and E is the set of all edges in *MI-Tree*.

We define a set of constraints for a *MI-Tree*:

- $N_C = \{C_i, i=1,...n\}, \forall C_i \in N_c \rightarrow C_i$ is an non-terminal node in the tree
- $C_i = \{d_1,....,d_{in},\}$, d_i is a representive point of cluster, n is the number of representive point of C_i
- $N_d = \{D_i \mid i=1,...,m\}, \forall D_i \in N_d \rightarrow D_i$ is a terminal node in the tree and it's parent node$\in N_C$
- $N_a = \{D_i \mid i=1,...,l\}, \forall D_i \in N_d \rightarrow D_i$ is a terminal node in the tree and it's parent node is the root

A sample *MI-Tree* is shown in Fig. 1, C_1 is a non-terminal node, represented a cluster, and D_{c4} is a terminal node represented a document clustered to C_3, and D_{a1} is a terminal node, signed as an isolated document.

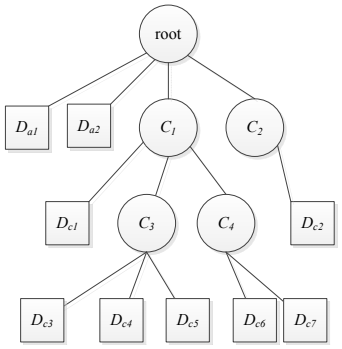


Figure 1. A sample of *MI-Tree*

C. Representative Points Selection

Ref. [5] uses the center as the representation point of a cluster, so the cluster covers spherical area. We suppose such a case: a cluster C contains only one document A, and some new documents are inserted into this cluster. If those new documents are more similar to each other than A, then the center of the cluster will depart from A (shown as Fig. 2).

When a new document B is coming and sim(A,B) is very high, $S_{BA} \approx 1$, B is supposed to be in same cluster of A. But the distance of B and C, sim(B,C) is smaller than the threshold value, B is excluded from this cluster.

This example indicates that index-tree is sensitive to the order of input data. It is partial to spherical data. To solve this problem, we propose the multi-presentation data to present a node in the tree.

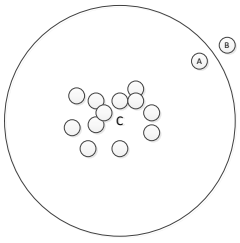


Figure 2. A sample of *MI-Tree*

The representative point of terminal node is itself.

For non-terminal node, with a subset of objects (C_i), the node can be represented (or typified) by the representative points (d_r). From Fig. 3, we can learn that nodes d_{ri} (i=1, 2, 3) are representative points of the Cluster, C is the center of the Cluster.

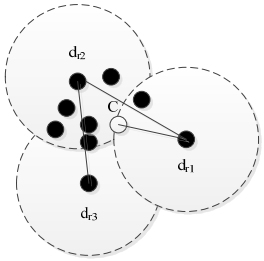


Figure 3. Multi-representation of non-terminal node

Selection can be achieved by picking up the farthest document to the previous representative points or the center of clusters to represent individual clusters the selection process can be formalized as:

$$\max_i \{| C_i - d_r \cup d_c |\} \quad .$$

Fig. 4 gives the pseudocode of representative points selection.

```
Function SelectMultiReprestation(C, REPREMAX)
 // C: a cluster
 // REPREMAX: the maximum number of representative points
  M = the center of C;
  For i=0 to REPREMAX
    FOREACH node IN c
      IF i==0 THEN
        get representative point which is farthest from center C
      ELSE
        get representative point which is farthest from representative
  points of C
      END IF
    NEXT
    Represents+=represent
  NEXT
  Return Represents
 END FUNCTION
```

Figure 4. Multi-representation selection algorithm

D. Similarity Calculation

The cosine between two document vectors is used to compare documents similarity. To prevent longer documents from dominating centroid calculations, normalizing all document vectors to unit length is needed.

As to two document d and d' at time t, their similarity is calculated as:

$$similarity_t(d,d') = \sum_{w \in d \cap d'} weight_t(d,w) * weight_t(d',w) \quad (4)$$

For two clusters C_x, C_y, the similarity is calculated as the arithmetic mean of similarity between representative points.

$$Sima(C_x, C_y) = \sum_{\forall D_m \in C_x, D_n \in C_y} sim(D_m, D_n) / M \times N \quad (5)$$

The maximum similarities between representations are also calculated for the later *MI-Tree* merging.

$$Simx(C_x, C_y) = Max(sim(D_m, D_n), \forall D_m \in C_i, D_n \in C_j) \quad (6)$$

We get the similarity between documents and clusters by calculating the shortest distance between documents and representative points in clusters using a cosine measurement (7)

$$Sim(d, C_r) = \min(similarity(d, d_{ri})) \quad (7)$$

To determine whether a document can be attributed to an old cluster or not, we gives several definitions:

Definition 1. Let d_C be a terminal node. Given a new point A, let s be the distance from A to C. A is said to form a higher dense region in d_C if $d>\theta$. (θ is user-defined)

Definition 2. Let d_r be a multi-representation of non-terminal node C. Given an upper limit $U_L = \max_{d_r \in C}\{similarity(d_r, d_c)\}$ (d_c is the center of the cluster C), the cluster C is homogeneous only if $d_i \leq U_L$ for $\forall d_i \in C$.

Definition 3. Let C be a non-terminal node. Given a new point A, let B be a representative points of C's cluster which is the nearest neighbor to A. Let d be the distance from A to B. A is said to form a higher dense region in C if $s>L_L$

In Fig. 5, because $s_{EC}<L_L$, so E will be added to cluster and D will be out of cluster because $s_{DC}>L_L$.

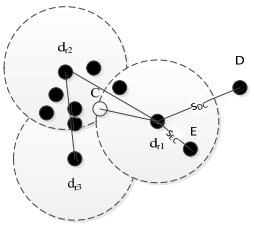


Figure 5. Relationship between a new document and a cluster

IV. New Cluster Detection Process

Traditional incremental methods updated the old clustering result for each new document. However, because of the document vector, some new documents reported related topics, are put into the different clusters. To avoid clustering sensitive to document input order, we deals the newly coming documents as a bulk, run the clustering process on this bulk first, then merge the old clustering tree into the new one.

For a new documents bulk, after the clustering process, a *MI-Tree* T_{new} is generated, then we merge the old one T_{old} into it, showed in Fig. 6 (The terminal nodes have been omitted).

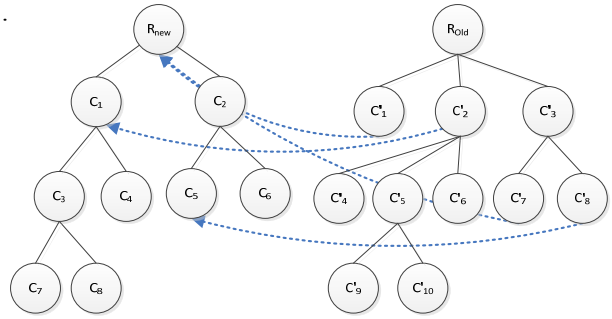


Figure 6. Example of a new tree and a old tree

To merge the two trees, we deal the cluster node set N_c first, then the terminal nodes linked directly to the root of the T_{new} and T_{old} (N_a)are processed.

The N_c nodes emerging process compare each cluster of the two trees with the depth first traversal to determine how to insert a cluster node of T_{old} into T_{new}. The detail steps are given as following:

Step 1: Calculated the similarity between each node located at first level in R_{old} with the peer nodes in R_{new}, e.g. the result of Fig. 3 is given in Table 1.

TABLE I. Similarity Between First Level Nodes

Similarity value	C'_1	C'_2	C'_3
C_1	0.32	0.56	0.21
C_2	0.28	0.35	0.43

Step 2: The pairs are picked out, if $Sima(C_i, C'_j) > \delta$, then the node C'_j is merged into C_i , e.g. the similarity between C_1 and C'_2 is 0.56, so repeating step 1 to 4 to merge them, as showed in Fig.3.

Step 3: If $Sima(C_i, C'_j) \leq \delta$, but $Simx(C_i, C'_j) > \delta$, e.g. C_2 and C'_3, the sub-clusters of them which pairs' similarity is larger than δ are picked, e.g. C_5 and C'_8, then the terminal documents of C'_8 are inserted into C_5 directly, and the representative points are renewed. C'_8 is deleted from C'_3, and the rest of C'_3 is inserted into R_{new} as sub-node.

Step 4: The rest nodes of R_{old} in level one are inserted into R_{new} as sub-node, e.g. C'_1, as showed in Fig. 3.

During the merging process, if any cluster node C_i of T_{new} hasn't been updated, then it is notified as a new cluster.

Each terminal node belonging to N_a (which is the document isolated to any clusters) is inserted to T_{new} one by one. The clusters generated on this step are also signed as new ones.

V. Experiment

A. Datasets and Experimental Setupf

We have constructed a text analysis platform aided with Lucence and Chinese lexical analysis tools. The

implemented model of detecting the new cluster is embedded into this platform to verify its quality.

We use three datasets to test our model. The documents of Dataset-1 are from learning channel of SOHU (http://learning.sohu.com) from June 1, 2008 to June 30, 2008, and Dataset-1 is from 2008 Olympic channel (http://2008.sohu.com). The documents of Dataset-3 are practical data from a business domain.

In our experiment, we iterate the clustering steps for every one day's data of Dataset-1 and Dataset-2, and process the Dataset-3 ten days' data one time increasingly. Table II illustrates the number of documents in each dataset and the new clusters' number which are labeled manually after first clustering iteration.

TABLE II. DATASETS FOR EXPERIMENTS

	Number of documents	*New events of dataset*
Dataset-1	3947	80%
Dataset-2	11629	87.16%
Dataset-3	5855	93.46%

B. *Clutering Accurracy*

We implemented System-1 based on dynamic indexing tree clustering [5] and System-2 with our approach presented based on MI-TREE.

CF- Feature [16] is a clustering quality evaluation method. Large CF-Feature is, better clustering result is.

We got largest CF-Feature when we set $\theta=0.5$ in train set, so we set $\theta=0.5$ in following experiments.

We evaluated both systems' clustering quality with three metrics: processing time, accuracy and CF-Feature.

The clustering accuracy is used as a measure of a clustering result. It is defined as :

$$\varphi = \sum_{i=1}^{k} x_i / N \qquad (8)$$

x_i is the number of object occurring in both the ith cluster and its corresponding class, N is the number of objects in the dataset. k is the resultant number of clustering. Table I shows the accuracy of system-1 and system-2.

TABLE III. ACCURACY ON SYSTEM-1 AND SYSTEM-2

Number of documents	accuracy	
	System-1	*System-2*
100	75%	80%
500	83.42%	87.16%
2000	89.46%	93.46%

Table II shows the CF-Feature of System-2 is large than that of System-1, when the number of documents exceeds 1000, so for large dataset, System-2 is better.

TABLE IV. CF-FEATURE BETWEEN CLUSTERS ON SYSTEM-1 AND SYSTEM-2

Number of documents	CF-Feature	
	System-1	*System-2*
100	0.50624	0.48285
500	0.40597	0.40129
1000	0.29695	0.29721
1500	0.27473	0.27669
2000	0.26211	0.26671

To process 2000 documents, system-1 and system-2 consume 455,522 and 219,995 milliseconds respectively, while we set REPREMAX as 5. It satisfies the theoretical analysis.

C. *Parameter Selection*

In our algorithm, we must customize two parameters to get better performance: MaxTreeLevel and δ.

The parameter MaxTreeLevel is taken to control the depth of the *MI-Tree*. We set MaxTreeLevel from 3 to 15 and run the algorithm on three datasets. Fig. 7 shows the relationship between MAXTREELEVEL and the number of discovered new clusters. The positions of the asterisks are the number of clusters selected manually of three datasets. It suggests that the new clusters detected are insensitive to this parameter. In programming, we set a smaller MaxTreeLevel as 5, it can save computational time.

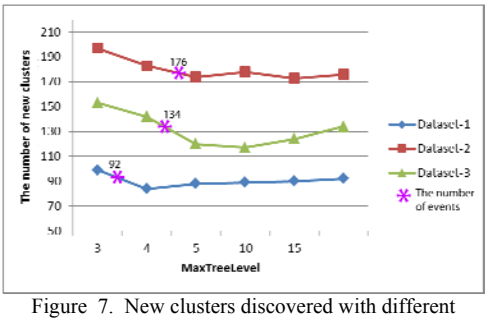


Figure 7. New clusters discovered with different MAXTREELEVEL

The parameter δ is used to measure the similarity between two clusters. Fig. 8 shows the relationship between δ and number of discovered new clusters.

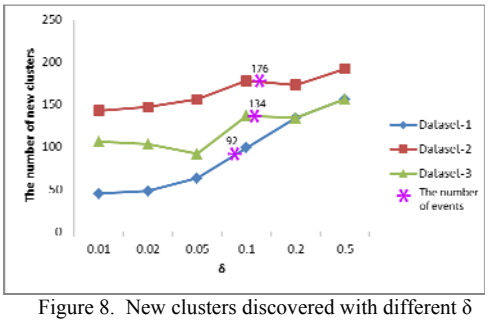


Figure 8. New clusters discovered with different δ

Fig. 8 shows the CF-Feature after clustered thirty times in three datasets, and different δ values. Accuracy is reducing, as δ is growing.

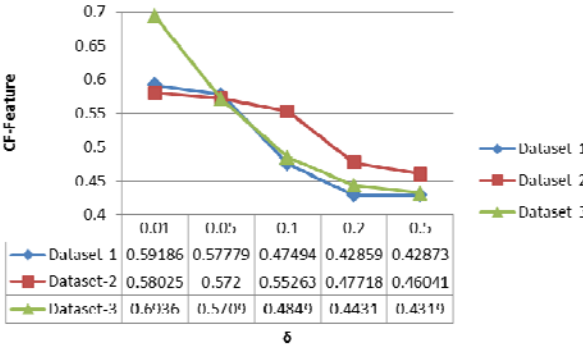


Figure 9. CF-Feature with different δ

Fig. 8 and Fig. 9 reveal: when δ>0.1, the number of new clusters is close to the expected number.

D. Experiment Result

We evaluated our system's quality in two aspects: one is the new clusters detected during the iterative clustering and merging process, the other is the accuracy of the clusters generated.

Fig. 10 is a part of new clusters generated after the clustering process on data in 6/7/2008 of dataset-2 (learning channel). The new clusters are some topics about "the college entrance exam". Though we get the cluster "entrance exam" on previous step, new sub clusters of different topics can be found and distinguished efficiently.

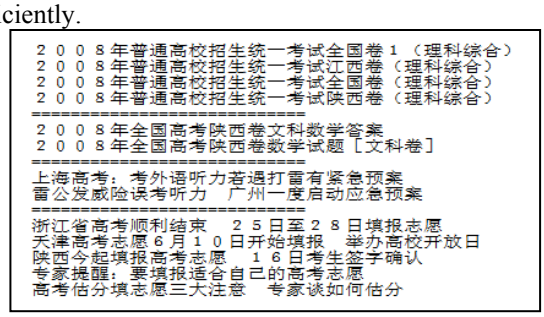
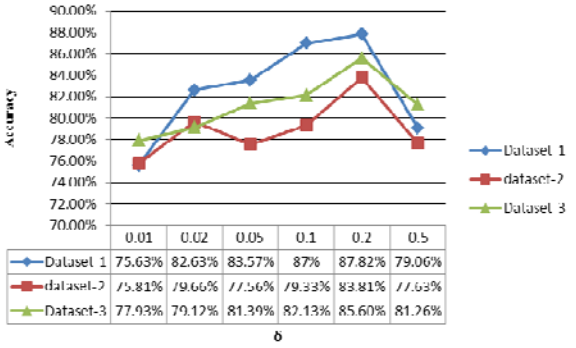


Figure 10. New clusters after 7th documents set is coming

The clustering accuracy is defined as (8). Fig. 11 shows the accuracy in three datasets with different δ.



	0.01	0.02	0.05	0.1	0.2	0.5
Dataset 1	75.63%	82.63%	83.57%	87%	87.82%	79.06%
dataset-2	75.81%	79.66%	77.56%	79.33%	83.81%	77.63%
Dataset-3	77.93%	79.12%	81.39%	82.13%	85.60%	81.26%

Figure11. Clustering accuracy with different δ

We found the algorithm can achieve best accuracy with δ setting as 0.2 , the accuracy on three datasets is as good as the state of the art of the clustering method.

VI. Conclusion

As the amount of generated information increases so rapidly in the digital world, valuable information mining becomes more and more important. In this paper, we tend to detect the trending events from online Web environment efficiently. Our work constructs a novel model based on MI-Tree text clustering. Instead of updating the clustering for each newly data, we process the new data for a period preferentially, and then compare the result with the old clustering to detect whether new clusters have generated. We test the model on Sogou dataset and business data, it achieves high accuracy, and the experiment on the practical application data shows our method can detect valuable information, and relieve people from time-consuming work of reviewing large amounts of documents manually. In the future, we will try to decompose the nodes whose representative points have become away from each other during the incremental iteration process, this can help in detecting new cluster more effectively.

Appendix A The English Version of Figure 10.

2008 National College Entrance Examination Volume 1 (Integrated Science)
2008 Jiangxi College Entrance Examination Study (integrated science)
2008 National College Entrance Examination Study (Integrated Science)
2008 Shaanxi College Entrance Examination Study (Integrated Science)
==========
2008 National College Entrance Mathematics for the Arts in Shaanxi answer paper
2008 National Study of Mathematics, Shaanxi College Entrance Examination [Study of Liberal Arts]
==========
Shanghai College Entrance Examination: Examination Listening In case of an emergency plan thunder
Thunder gotta play a dangerous mistake once in Guangzhou start hearing test emergency plans
==========
College Entrance Exam successful conclusion of the 25 to 28 volunteers fill
Tianjin College Entrance Examination on June 10 began to fill open day organized by University
Shaanxi for managing volunteers in completing college entrance examination confirmed the signing on 16
Experts advise: to fill in for their College Entrance Examination
Entrance Gufen fill volunteer experts discussing how the three note Gufen

Acknowledgment

This work was supported in part by a grant from the National Natural Science Foundation of China (No. 60903160).

References

[1] Oren Zamir, Oren Etzioni, "Web Document Clustering: A Feasibility Demonstration", in *Proceedings of SIGIR'98*, Melbourne, Australia, 1998.

[2] Nachiketa Sahoo , Jamie Callan , Ramayya Krishnan , George Duncan , Rema Padman, "Incremental hierarchical clustering of text documents", in *Proceedings of the 15th ACM international conference on Information and knowledge management*, November, 2006, Arlington, Virginia, USA.

[3] B. Thorsten, C. Francine, and F. Ayman. "A System for New Event Detection". in *Proceedings of the 26th Annual International ACM SIGIR Conference*, pp:330–337, New York, NY, USA. 2003.

[4] T. Brants and F. Chen, “A system for new event detection”, In *Proceedings of SIGIR’03*, pp:330–337, Toronto, Canada, 2003. ACM.

[5] K. Zhang, J. Z. Li, and G. Wu, “New event detection based on indexing-tree and named entity”. In *Proceedings of SIGIR’07*, pp:215–222, Amsterdam, The Netherlands, 2007. ACM.

[6] Wim De Smet, Marie-Francine Moens, ”An Aspect Based Document Representation for Event Clustering”, in *Proceedings of the 19th Meeting of Computational Linguistics in the Netherlands*, pp:55-68, 2009.

[7] Gavin Shaw, Yue Xu, “Enhancing an Incremental Clustering Algorithm for Web Page Collections”, in *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, September, 2009.

[8] G.P.C. Fung, J.X. Yu, H. Liu and P.S. Yu. “Time-Dependent Event Hierarchy Construction”. in *Proceedings of KDD-2007*, pp 300-309, 2007.

[9] W. Wong and A. Fu, “Incremental document clustering for web page classification”, in *Proceedings of International Conference on Information Society*, Japan, 2000.

[10] M. Charikar, C. Chekuri, T. Feder, and R. Motwani, “Incremental clustering and dynamic information retrieval”, in *The 29th annual ACM symposium on Theory of computing*, pp:626-635, 1997.

[11] K. Hammouda and M. Kamel, “Incremental document clustering using cluster similarity histograms”, in *IEEE/WIC International Conference on Web Intelligence*, 2003.

[12] Chung-Chian Hsu, Yan-Ping Huang. “Incremental clustering of mixed data based on distance hierarchy”. in *Expert Systems with Applications*, vol(35), pp:1177– 1185, 2008.

[13] Maria Soledad Pera , Yiu-Kai Ng, “Utilizing phrase-similarity measures for detecting and clustering informative RSS news articles”, *Integrated Computer-Aided Engineering*, vol.15, pp.331-350, December 2008.

[14] Weimao Ke, Cassidy R. Sugimoto, Javed Mostafa, “Dynamicity vs. effectiveness: studying online clustering for scatter/gather”, in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, July 19-23, 2009.

[15] Canhui Wang, Min Zhang, Shaoping Ma, Liyun Ru, “Automatic online news issue construction in web environment”, in *Proceeding of the 17th international conference on World Wide Web*, April 21-25, 2008.

[16] Han xi-wu, Zhao Tie-jun. “An evaluation method for clustering quality and its application,” *Journal of harbin institute of technology*, vol 41, pp.225-227, November 2009, 225-227.(In Chinese)

Hui Song was born in Hubei China， on February 17,1971. She received the B.S. degree in Computing Mathematics from Xi’an Jiaotong University in 1995 and the Ph.D degree in System Architecture of Computer Science from Shanghai Jiaotong University in 2004.

She is an associated professor working at Donghua University , Shanghai China. Her research interests include Web information mining, information processing intelligence and information integration.

Lifeng Wang was born in ShanXi China， on June 8,1986. He received the B.S. degree in Computer Software from Donghua University in 2011. He research interests is Web information mining.

Baiyan Li is born was LiaoNing China，on August 23,1968. He received the B.S. degree in Computer Science from Nanchang University in 1993 and the Ph.D degree in System Architecture of Computer Science from Shanghai Jiaotong University in 2005.

He is an associated professor working at Donghua University, Shanghai China. His research interests include pattern recognition, parallel processing.

Xiaoqiang Liu was born in Henongjiang China， on September 24,1968. She received the B.S. degree in Computer Science from Harbin Institute of Technology in 1993 and the Ph.D degree in Computer software from Donghua University in 2005.

She is a professor working at Donghua University, Shanghai China. Her research interests include knowledge management and knowledge engineering.

