Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao, 11-14 July 2010

ONLINE TOPIC DETECTION AND TRACKING OF FINANCIAL NEWS BASED ON HIERARCHICAL CLUSTERING

XIANG-YING DAI, QING-CAI CHEN, XIAO-LONG WANG, JUN XU

Intelligence Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China
E-MAIL: michealdai@gmail.com, qingcai.chen@gmail.com, wangxl@insun.hit.edu.cn, hit.xujun@gmail.com

Abstract:

In this paper, we apply TDT technology to the vertical search engine in the financial field. The returned results are grouped into several topics with the stock as the unit. Then we show the topics to the users in time series order. As a result, users can easily learn about the important events which belong to a stock. Moreover, the causes and the effects of these events can also be found out easily. We improve the common agglomerative hierarchical clustering algorithm based on average-link method, which is then used to implement the retrospective topic detection and the online topic detection of news stories of the stocks. Additionally, the improved single pass clustering algorithm is employed to accomplish topic tracking. We consider that the feature terms which occur in the title of a news story contribute more during the similarity calculation and increase their corresponding weights. Experiments are performed on two datasets which are annotated by human judgment. The results show that the proposed method can effectively detect and track the online financial topics.

Keywords:

Topic Detection and Tracking; Agglomerative Hierarchical Clustering; Vector Space Model

1. Introduction

For the financial information service, users hope that they can hold the important events of a stock as well as the causes and the effects of these events. There exist several drawbacks to the finance portals, such as the coverage of a stock's news stories is not extensive and news stories which belong to different stocks are mixed together. Meanwhile, massive similar and follow-up news stories about the same event are published via different news media. What's worse, the reprints of web pages cause that there exit a large quantity of repeated news stories. All of the deficiencies make it hard to search the related news stories of a stock by browsing finance portals. The vertical search engine in the financial field, such as Google Finance, can provide browsing service of news stories by stock for users. However, the results are not organized by topic and time. As a result, it is not convenient for users to browse the first story of an event and track the causes and the effects of the event. Therefore, how to detect and track the important events from the searched results, then showing these events to the users in time series order and in the form of topics will be the next problem to be resolved for the vertical search in the financial field.

In this paper, we use the topic detection and tracking (TDT) technology to solve the above problem. TDT is a kind of technology which can organize the news stories into the news topics. All the news stories come from news story streams. A topic consists of many news stories related to it. These related stories cover from the initial news story to the follow-up news stories. In order to accomplish this task, we need to do the following jobs. Firstly, it is necessary to detect the existing topics from the previous story collection before running the online topic detection and tracking in the system. We call this procedure retrospective topic detection. Secondly, we need to process the online news stories as they arise for the detection of the new topics contained in them. This is the implementation of the online topic detection. Lastly, when the news stories arise, we process them immediately to decide whether the news stories are the related ones of the previous topics. We refer to this procedure as the implementation of the online topic tracking. The followings are the corresponding examples. The detection of the topic "China CITIC Bank had bought a 70.32% stake in CITIC International Financial Holdings Ltd." from the previous story collection is an example of the retrospective topic detection. Detecting a new topic "China CITIC Bank promote sound financial plan" from the follow-up stories belongs to the online topic detection. Finally, "China CITIC Bank had bought a 70.32% stake in CITIC International Financial Holdings Ltd., and kicked off the process of internationalization" is a story which is related to the topic "China CITIC Bank had bought a 70.32% stake in CITIC International Financial Holdings Ltd.", and the detection of the story from the follow-up stories is an example of the topic

978-1-4244-6527-9/10/$26.00 ©2010 IEEE

tracking.

The rest of the paper is organized as follows. In section 2, we discuss the related work of TDT. In section 3, we improve the agglomerative hierarchical clustering method, and apply it to the retrospective topic detection and the online topic detection. Additionally, to implement topic tracking, we employ the improved single pass clustering method. Section 4 describes the corpus used in our experiments and the evaluation metrics. In section 5, we present our experiments and the results while section 6 concludes the paper with the observations on our results.

2. Related work

TDT contains two main subtasks, topic detection and topic tracking. The topic detection consists of the retrospective topic detection and the online topic detection. The main technologies used in these tasks are the agglomerative hierarchical clustering method [1, 2] and the single pass clustering method [1, 3, 4, 5].

For the retrospective topic detection, we need to find out the topics that exist in the previous story collection, many works have already been done for this task. In 1998, Yang of CMU proposed the retrospective event detection which is a study of the retrospective topic detection [1]. Moreover, considering more on modeling events in probabilistic manners as well as the better representations of articles and events, Li proposed a probabilistic method to incorporate both content and time information in a unified framework [6].

For the online topic detection, the new topics should be detected from the follow-up news stories. Most of the researchers focus on the selection of the clustering methods. Furthermore, single-pass clustering method is used widely [1, 3].

In the topic tracking task, the related stories of the previous topics should be detected from the follow-up news stories. In UMASS's method [7], the relevance of the current topic model and the follow-up news stories are computed based on the statistical methods. Then they recognize the related story by the relevance and combine it into the corresponding topic. Lastly, the topic model is rebuilt.

3. Topic detection and tracking based on clustering

We implement the retrospective topic detection and the online topic detection independently, and the retrospective topic detection is the infrastructure of the online topic detection. Firstly, the improved agglomerative hierarchical clustering method is used to detect topics from the previous story collection. Secondly, considering the phenomenon that most of stories about one certain topic usually burst in a short period of time, we adopt the improved agglomerative hierarchical clustering method to generate the set of candidate topics by clustering stories which occur in a short time interval Δ_t, and during this process, the average-link method is used to compute similarity between topics. Then, the incremental clustering method [9] is employed to process the candidate topics one by one. If the similarity between a candidate topic and each single previous topic within the latest period of time Δ_T is smaller than the threshold θ_n, we consider that a new topic occurs.

3.1. Pre-process and story representation

For keeping in step with TDT, we think the title part and the content part of a web page constitute a story.

During preprocessing, the first step is to remove the redundant stories. Secondly, word segmentation is performed. After the two steps, we remove the stopwords. It is important to point out that not only the common stopwords but also the special stopwords in the financial field are removed. At last, we use the vector space model to represent a story. That is to say, a story is represented as a term vector.

In the related research of the retrospective topic detection [1], TF-IDF model was employed to compute the feature weight. Besides, the incremental TF-IDF model was adopted to do the same task in much online topic detection research [1, 3, 4]. Due to the good performance the two models have achieved in our experiments, we use TF-IDF model in the retrospective topic detection, whereas the incremental TF-IDF model is employed for implementing the online topic detection. Meanwhile, during calculating the feature weight, we assign higher weight to the words which occur in the title part of the story.

In the retrospective topic detection, feature weight is calculated using TF-IDF model. Then we normalized the gained feature vector. TF (term frequency) is calculated as follows:

$$tf(t,d) = \begin{cases} \alpha * TF(t,d) & \text{if } t \in T \\ TF(t,d) & \text{if } t \notin T \end{cases} \quad (1)$$

where $TF(t,d)$ means the number of times term t occurs in the story d, T represents the term set which consist of terms occurring in the title part of the story, and α is the weight coefficient. In our experiments, we find the system can achieve the best performance when α is set to 1.4. Formula (2) is used to calculate the feature weight, where N denotes the whole number of stories contained in the corpus used in the retrospective topic detection, while df_t represents the number of stories in which term t occurs.

$$weight(t,d)=\frac{tf(t,d)*\log((N+1)/(df_t+0.5))}{\sqrt{\sum_{t'\in d}(tf(t',d)*\log((N+1)/(df_{t'}+0.5)))^2}} \quad (2)$$

During the implementation of the online topic detection, we make use of the incremental TF-IDF model to calculate the feature weight and then normalize the obtained feature vector. In this part, after processing a news story, we need to update the inverse document frequency (IDF) immediately. The IDF in the incremental TF-IDF model is calculated as follows:

$$idf(t,c)=\log(N_c/n(t,c)) \quad (3)$$

where c represents the current time, N_c means the total number of stories at the current time c, $n(t,c)$ denotes the total number of stories which contain term t at the current time c. In our system, we run our system every half a day, hence, we can determine value of c. The following formula is used to calculate the feature weight.

$$weight(t,d)=\frac{tf(t,d)*idf(t,c)}{\sqrt{\sum_{t'\in d}(tf(t',d)*idf(t',c))^2}} \quad (4)$$

3.2. Similarity calculation

In this paper, we use cosine similarity to calculate the similarity between two stories. For instance, similarity between story d_1 and d_2 is calculated as follows:

$$similarity(d_1,d_2)=\frac{\sum_{t\in d_1\cap d_2}weight(t,d_1)*weight(t,d_2)}{\sqrt{\sum_{t_1\in d_1}weight(t_1,d_1)^2}*\sqrt{\sum_{t_2\in d_2}weight(t_2,d_2)^2}} \quad (5)$$

3.3. Retrospective topic detection

The agglomerative hierarchical clustering is a kind of bottom-up clustering method. We view each story as a topic at the beginning, and then enter into the following iterations. In each iteration, we combine the two most similar topics into a single topic firstly, and then recalculate the similarity between the new topic and the other topics. The iterations are performed until the maximum similarity is smaller than the predefined threshold. In this paper, we improve the group-average agglomerative hierarchical clustering algorithm [1] through splitting the original algorithm into the following two steps. In the first step, we calculate the similarity of each pair of two topics, and directly combine the two topics if the similarity between them is higher than some threshold θ_1. Then we rebuild the topic model. In the second step, we perform the universal agglomerative hierarchical clustering algorithm. In our experiments, θ_1 is determined empirically, the system achieves the best performance when θ_1 is set to 0.7.

The procedure of clustering is described as follows:

1. Preprocess all stories, and represent each of them by a feature vector, then view each story as a topic.
2. Let <similarity，<topic1, topic2>> denote a two-tuple of two different topics and their similarity. After calculating the similarity of each pair of two different topics, generate a list of two-tuples in descending order of the calculated similarity.
3. For those two-tuples whose corresponding similarity is higher than the threshold θ_1, process each of them by combining the corresponding two topics into a new topic.
4. Rebuild the topic model for the new topics, which are used to replace the original topics. Finally, recalculate the similarity of each pair of two different topics and regenerate the list of two-tuples in descending order of the calculated similarity.
5. If the similarity of the first two-tuple is higher than the threshold θ_2 ($\theta_1 > \theta_2$), combine the corresponding two topics into a new topic. Then replace the two topics with the new topic, and go to the step 4. Otherwise, the algorithm terminates.

3.4. Online topic detection

In this section, the improved single-pass clustering method is used to implement our system. The process of clustering is described as follows:

1. Process the stories every time interval Δ_t using the improved agglomerative hierarchical clustering algorithm, and all those processed stories are the ones which arise in Δ_t. We can get the set of candidate topics, i.e. CTS.
2. Get a topic *ct* from CTS, and calculate the similarity between *ct* and each single previous topic within the latest period of time Δ_T. If the maximum similarity is smaller than the threshold θ_n, we consider *ct* a new topic.
3. Delete *ct* from CTS, if CTS is empty, then the algorithm terminates. Otherwise, the algorithm goes to step 2.

3.5. Topic tracking based on clustering

The topic tracking and online topic detection are performed simultaneously in our system. The process of clustering used in the topic tracking is described as follows:

1. Process the stories every time interval Δ_t using the improved agglomerative hierarchical clustering

algorithm, and all those processed stories are the ones which arise in Δ_t. We can get the set of candidate topics, i.e. CTS.

2. Get a topic *ct* from CTS, and calculate the similarity between *ct* and each single previous topic within the latest period of time Δ_T. If the maximum similarity, which is the similarity between *ct* and the previous topic T_c, is not smaller than the threshold θ_n, we consider that *ct* is related to T_c.
3. Combine the topic *ct* into the previous topic T_c, and rebuild the topic model.
4. Delete *ct* from CTS, if CTS is empty, then the algorithm terminates. Otherwise, the algorithm goes to step 2.

4. Dataset and evaluation metrics

4.1. Dataset

Our experiments are performed on two Chinese datasets, which are constructed from web pages downloaded from various financial portals. We annotate only the web pages which are published from June 1 to Aug 31 in 2009. The part of the corpus from June 1 to July 31 in 2009 is used to perform the retrospective topic detection, while the rest is employed to perform the online topic detection and tracking.

Dataset1: This dataset contains 471 news stories with 15 topics, and there are 11 topics contained in the part of corpus from June 1 to July 31 in 2009. In the rest of the corpus, 4 new topics are contained. The maximum topic has 93 news stories, while the minimum one has 11 news stories.

Dataset2: This dataset contains 1325 news stories with 29 topics, and there are 22 topics contained in the part of corpus from June 1 to July 31 in 2009. In the rest of the corpus, 7 new topics are contained. The maximum topic has 367 news stories, while the minimum one has 9 news stories.

4.2. Evaluation metric

In this paper, we adopt the traditional evaluation metrics which are widely used in clustering and Information Retrieval [10]: Recall, Precision and F-measure. In addition, we don't evaluate each part of the system but view the three parts of the system as a whole. Then we calculate the Precision, Recall and F-measure on the whole corpus to observe the global performance of the system.

We call the topic which is generated by our system a cluster, while the actual topic which is created by manual annotation is referred to as a class. And Precision, Recall and F-measure are calculated by [11]:

$$\Pr ecision(i,j) = \frac{n_{ij}}{n_j} \tag{7}$$

$$\mathrm{Re}call(i,j) = \frac{n_{ij}}{n_i} \tag{8}$$

where n_i and n_j are the sizes of class i and cluster j, respectively. n_{ij} denotes the number of members of class i in cluster j. Then, the F-measure of cluster j and class i is defined by [11]:

$$F(i,j) = \frac{2 * \mathrm{Re}call(i,j) * \Pr ecision(i,j)}{\mathrm{Re}call(i,j) + \Pr ecision(i,j)} \tag{9}$$

F-measure of each class i is defined as:

$$F_i = \arg\max_j (F(i,j)) \tag{10}$$

Similarly, the Precision and Recall of class i are defined as the corresponding values. We call Precision, Recall and F-measure of the class i P_i, R_i and F_i, respectively.

We can get the global Precision, Recall and F-measure by calculating the weighted mean value for the corresponding metric as follows [11]:

$$\Pr ecision = \sum_i \frac{n_i}{n} P_i \tag{11}$$

$$\mathrm{Re}call = \sum_i \frac{n_i}{n} R_i \tag{12}$$

$$F = \sum_i \frac{n_i}{n} F_i \tag{13}$$

where n is the total number of stories in the corpus.

5. Experiments and discussion

5.1. Experiment results

Our algorithm is based on the agglomerative hierarchical clustering (AHC) method, which is improved by bringing in two-step operation. And we use average-link method to compute the similarity between topics. We consider the importance of the title part in a story. That is if a term occurs in the title, it will be assigned higher weight. In our experiments, we compare the performance of the original algorithm and the improved one under different weight coefficients on dataset1 and dataset2, respectively.

TABLE 1. PERFORMANCE COMPARISON BETWEEN TWO ALGORITHMS ON DATASET1 (WEIGHT COEFFICIENTS OF TITLE WORDS ARE SET TO 1).

Method	Precision	Recall	F-measure
AHC	95.63%	99.15%	97.29%
Imp-AHC	95.63%	99.15%	97.29%

TABLE 2. PERFORMANCE COMPARISON BETWEEN TWO ALGORITHMS ON DATASET1 (WEIGHT COEFFICIENTS OF TITLE WORDS ARE SET TO 1.4).

Method	Precision	Recall	F-measure
AHC	95.65%	99.36%	97.40%
Imp-AHC	95.83%	99.15%	97.40%

TABLE 3. PERFORMANCE COMPARISON OF AHC UNDER TWO DIFFERENT WEIGHT COEFFICIENTS OF TITLE WORDS ON DATASET1.

Coefficient	Precision	Recall	F-measure
α =1	95.63%	99.15%	97.29%
α =1.4	95.65%	99.36%	97.40%

TABLE 4. PERFORMANCE COMPARISON OF IMPROVED AHC UNDER TWO DIFFERENT WEIGHT COEFFICIENTS OF TITLE WORDS ON DATASET1.

Coefficient	Precision	Recall	F-measure
α =1	95.63%	99.15%	97.29%
α =1.4	95.83%	99.15%	97.40%

TABLE 5. PERFORMANCE COMPARISON BETWEEN TWO ALGORITHMS ON DATASET2 (WEIGHT COEFFICIENTS OF TITLE WORDS ARE SET TO 1).

Method	Precision	Recall	F-measure
AHC	89.39%	93.43%	90.40%
Imp-AHC	89.48%	93.43%	90.43%

TABLE 6. PERFORMANCE COMPARISON BETWEEN TWO ALGORITHMS ON DATASET2 (WEIGHT COEFFICIENTS OF TITLE WORDS ARE SET TO 1.4).

Method	Precision	Recall	F-measure
AHC	90.46%	93.66%	90.99%
Imp-AHC	89.54%	96.22%	92.06%

TABLE 7. PERFORMANCE COMPARISON OF AHC UNDER TWO DIFFERENT WEIGHT COEFFICIENTS OF TITLE WORDS ON DATASET2.

Coefficient	Precision	Recall	F-measure
α =1	89.39%	93.43%	90.40%
α =1.4	90.46%	93.66%	90.99%

TABLE 8. PERFORMANCE COMPARISON OF IMPROVED AHC UNDER TWO DIFFERENT WEIGHT COEFFICIENTS OF TITLE WORDS ON DATASET2.

Coefficient	Precision	Recall	F-measure
α =1	89.48%	93.43%	90.43%
α =1.4	89.54%	96.22%	92.06%

Tables 1-8 describe the performance of the two algorithms under different weight coefficients of title terms on dataset1 and dataset2. From these tables, we can see that on dataset1, the Precision, Recall and F-measure run up to 95.6%, 99.1% and 97.2%, respectively. And on dataset2, the three corresponding values are as high as 89.3%, 93.4%, 90.4%. The performance is good enough to satisfy the need of our application.

5.2. Results analysis

From Table3, Table4, Table7 and Table8, we can find out that Precision, Recall and F-measure are all improved when we assign terms which occur in the title part much higher weight. The reason is that the title part of the story is the simplification of the content part, therefore terms which occur in the title is more representative. Those terms occur in the content part frequently, and most of them are the members of the feature vector. So, we can improve the system performance by increasing the weight of these terms.

Although the improved agglomerative hierarchical clustering algorithm doesn't improve the performance of system on dataset1, it does improve the performance on the dataset2. Table 5 and Table 6 illustrate this obviously. The algorithm directly combines the two topics between which the similarity is higher than θ_1. This is helpful to create the better topic model. When the size of the corpus is small, we can't observe this advantage. However, if the size is big enough, especially when there are a large number of stories in one topic, we can get better performance by applying the improved algorithm.

6. Conclusions

In this paper, an improved agglomerative hierarchical clustering method is proposed, which is applied to the topic detection of financial news. What's more, we implement retrospective topic detection, online topic detection and topic tracking with this method, respectively.

First, we split the original agglomerative hierarchical clustering method into two steps and consider the time factor. Then we compare a candidate topic only with the previous topics occur in the latest time interval Δ_T. Second, we view the title and the content as a whole story, and consider assigning different weight to the title terms during the similarity calculation [2]. Then we study the effect on the similarity between two stories.

The experiment results show that our method performs well on the online topic detection and tracking of financial news.

Acknowledgements

This work is supported by Natural Scientific Research Innovation Foundation in China (No. 6070301 and No. 60973076).

References

[1] Y. M. Yang, T. Pierce, J. Carbonell. A Study on Retrospective and On-Line event detection. *Proceedings of the 21st Annual International ACM SIGIR Conference, Melbourne, Australia*. pp. 37-45, 1998.

[2] C. H. Wang, M. Zhang, S. P. Ma, et al. Automatic online news issue construction in web environment. *Proceeding of the 17th International Conference on World Wide Web*, *Beijing,China*, pp. 21-25, April 2008.

[3] J. Allan, R. Papka, V. Lavrenko V. On-Line New Event Detection and Tracking. *Proceedings of the 21st Annual International ACM SIGIR Conference, Melbourne, Australia*, pp. 37-45, 1998.

[4] B. Thorsten, C. Francine, F. Ayman. A System for New Event Detection. *Proceedings of the 26th Annual International ACM SIGIR Conference, New York, USA*, pp. 330-337, 2003

[5] G. Kumaran, J. Allan. Text Classification and Named Entities for New Event Detection. *Proceedings of the 27th Annual International ACM SIGIR Conference, Sheffield, UK,* pp. 297-304, 2004

[6] Z. W. Li, B. Wang, M. J. Li, W. Y. Ma. A Probabilistic Model for Retrospective News Event Detection. *Proceedings of the 28th Annual International ACM SIGIR,* Salvador, Brazil, pp. 06-113, 2005.

[7] J. Allan, V. Lavrenko, D. Frey, V. Khandelwal. UMass at TDT 2000. *Proceedings of Topic Detection and Tracking Workshop*, 2000.

[8] Y. Hong, Y. Zhang, T. Liu, S. Li. Topic Detection and Tracking Review. *Journal of Chinese Information Processing ,* 21(6):71-87, 2007.

[9] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, R. Padman. Incremental hierarchical clustering of text documents. *Proceedings of the 15th ACM International Conference on Information and knowledge management, Arlington， Virginia，USA,* pp. 06-11, November 2006.

[10] C. J. van Rijsbergen. *Information Retrieval, second edition*. Buttersworth, London, 1989.

[11] M. Steinbach, G. Karypis, V. Kumar. A comparison of document clustering techniques. *KDD-2000 Workshop on Text Mining, Boston，MA，USA*, pp. 1-20, August 2000.