A Novel Technique of Topic Detection for On-line Text Documents: A Topic Tree-based Approach

Man Xuan, Han-Joon Kim*
School of Electrical and Computer Engineering, University of Seoul
e-mail : mmxuan23@gmail.com, khj@uos.ac.kr

온라인 텍스트문서의 계층적 트리 기반 주제탐색 기법

현만, 김한준*
서울시립대학교 전자전기컴퓨터공학부

Abstract

Topic detection is a problem of discovering the topics of online publishing documents. For topic detection, it is important to extract correct topic words and to show the topical words easily to understand. We consider a topic tree-based approach to more effectively and more briefly show the result of topic detection for online text documents. In this paper, to achieve the topic tree-based topic detection, we propose a new term weighting method, called CTF-CDF-IDF, which is simple yet effective. Moreover, we have modified a conventional clustering method, which we call incremental k-medoids algorithm. Our experimental results with Reuters-21578 and Google news collections show that the proposed method is very useful for topic detection.

1. Introduction

In this paper, we consider the problem of discovering significant topics from incoming online publishing documents and showing their change. We collect the increasing documents as time goes on, and build topic trees every specific time. The topic tree is a sort of hierarchical structure with topic words. The root of a topic tree is a general topic word, the internal nodes and leaves of a topic tree correspond to more specific topic words. It helps users to grasp the main content easily.

To achieve the above goal, we propose a novel technique of topic detection, which automatically generate a topic tree structure. The topic tree can be developed by computing so called 'subsumption relationships' among topic words.

The topic trees are built through the following 3 steps: the first step is to partition incoming documents by an incremental clustering method (cf. Section 2), and the second one is to extract significant topic words from each of clusters (cf. Section 3). The final step is to developing subsumption relationships among topic words (cf. Section 4).

2. Incremental Clustering Method for Topic Detection

In our work, we intend to use clustering techniques in order to detect topics from incoming online documents. We note that the clustering process for topic detection should be performed for streaming documents; thus the incremental clustering method is suitable for topic detection.

The incremental clustering is a dynamic clustering technique that can continuously decompose newly incoming documents while inserting them into the current clusters.

One of popular incremental clustering methods is the incremental k-means algorithm [1]. In our work, we have modified the conventional 'incremental k-means algorithm', which we call 'incremental k-medoids algorithm'. The proposed algorithm is more robust to noise and outliers. And we use a new method to decide whether to merge a new document with a cluster or allow the document to be a seed for a new cluster. We simply use TF (Term Frequency) to weigh terms and use the cosine similarity to evaluate similarities between documents when clustering documents.

2.1 Incremental k-medoids Algorithm

The proposed incremental k-medoids algorithm is an algorithm that combines the conventional incremental clustering algorithm [1] and the k-medoids clustering algorithm [2]. Whenever there is a new incoming document, the incremental k-medoids algorithm runs as the following procedure:

1. Search for the cluster closest to the incoming document, and calculate the similarity value between them.
2. Compare the similarity value obtained in Step 1 with the given threshold (discussed in Section 1.2) to decide whether to merge the document with the cluster or set the document as a seed of a new cluster.
3. Update the medoids.
4. For each document, merge it with the cluster closet to it.
5. Repeat Step 3 and Step 4 until the clustering result

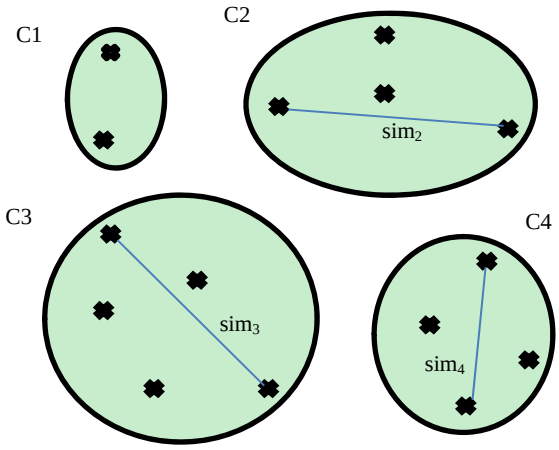
* Corresponding author: Han-Joon Kim (khj@uos.ac.kr)

does not change.

Here, the similarity value between a document and a cluster is the similarity value between the document and the cluster's medoids. If a cluster is the cluster closet to a document, it means the similarity value between the cluster and the document is larger than the similarity value between any other cluster and the document.

2.2 Threshold for Incremental Clustering

For incremental clustering, we need to give a particular threshold value in order to decide whether to merge each of incoming documents with its closest cluster or to allow the document to be a seed of a new cluster. The threshold value allows the incremental clustering process to determine the size of clusters. Since topic trees are built from each cluster, the threshold value determines how general (or specific) the topic trees will be. In practice, it is desirable to automatically adjust the threshold value.



Step 1: Select C2, C3, C4;
Step 2: Calculate sim_2, sim_3, sim_4;
Step 3: Calculate the threshold:
threshold $= \frac{sim_2+sim_3+sim_4}{3}$.

(Figure 1) The incremental clustering process

At the initial step, several clusters are given as initial clusters, and calculate the threshold value as follows. The procedure is illustrated in Figure 1.

1. Select the clusters that have more than three documents.
2. Calculate the similarities between the farthest documents in each of selected clusters in Step 1.
3. Calculate the average of similarities calculated in Step 2. This average is used as a threshold value.

Whenever new documents are inserted in the system, it updates the threshold value through the above procedure.

As discussed above, the initial threshold value depends on the initial clusters. And the initial threshold value influence the size of clusters being clustered. Hence, to get a better result, it is necessary to setup several initial clusters manually. Setup the initial clusters with documents that similar to each other appropriately. And the initial clusters should be distinct from each cluster.

3. Extracting Topic Words from a Set of Clusters

Commonly, the topic words which can describe a cluster are the words that appear frequently (which means popularity) in the cluster but infrequently (which means particular) overall. To isolate such topic words, we propose three measures: Cluster Term Frequency (CTF), Cluster Document Frequency (CDF), and Inverse Cluster Frequency (ICF). Actually, these measures are based on the idea of conventional IR weighting measures such as TF, DF, and IDF [3].

3.1 Cluster Term Frequency (CTF)

Firstly, we define the cluster term frequency (CTF) of a given term in a cluster, which means the number of frequency which the given term appears in the cluster. It can indicate the popularity of the given term in the cluster. And, to prevent a bias towards length of documents, we need to normalize the CTF measure. Finally, the Cluster Term Frequency of term *t* in cluster *c* is calculated as follows:

$$ctf(t,\mathrm{c}) = \frac{tc(t,\mathrm{c})}{\sum_{i=1}^{n} tc(\mathrm{t_i},\mathrm{c})} \quad (1)$$

where $tc(t,c)$ is the term frequency of the term *t* in cluster *c*. $\sum_{i=1}^{n} tc(\mathrm{t_i},\mathrm{c})$ is the sum of term counts of all terms in cluster *c*. And, *n* is the number of terms in cluster *c*.

3.2 Cluster Document Frequency (CDF)

Also, we consider the document frequency in each cluster, which is called 'cluster document frequency (CDF)'. CDF of a given term in a cluster is the number of documents which have the given term in the cluster. It can indicate the popularity of the given term in documents of the cluster. And to remove a bias towards the number of documents, we need to normalize the CDF measure. Finally, the Cluster Document Frequency of term *t* in cluster *c* is calculated as follows:

$$cdf(t,\mathrm{c}) = \frac{dc(t,\mathrm{c})}{D} \quad (2)$$

where $dc(t,\mathrm{c})$ is the document frequency of the term *t* in cluster *c*. D is the total number of documents in cluster *c*.

3.3 Inverse Cluster Frequency (ICF)

Finally, we consider the number of clusters in which the given term occurs, which we call 'cluster frequency'. It indicates the popularity of the given term in whole clusters. To indicate the particular of the given term, we need to normalize the cluster frequency to the Inverse Cluster Frequency (ICF). Finally, the Inverse Cluster Frequency of term *t* is defined as follows:

$$icf(t) = \log_2 \frac{C+1}{cc(t)+1} \quad (3)$$

where $cc(t)$ is the cluster frequency of the term *t*, and C is the total number of clusters.

3.4 Combining the Measures

As mentioned above, the cluster term frequency and cluster document frequency can indicate the popularity of a given term in a cluster, and inverse cluster frequency can indicate the particular of the given term in whole clusters. Here, in order to express the descriptive power of a given term which can significantly describe a cluster, we combine the above three measures, which is denoted as CTF-CDF-ICF.

In cluster *c*, for every candidate topic word *t*, we calculate CTF-CDF-ICF as follows:

$$\text{CTF-CDF-ICF} = ctf(t,\text{c}) \cdot cdf(t,\text{c}) \cdot icf(t)$$

$$= \frac{tc(t,\text{c})}{\sum_{i=1}^{n} tc(\text{t}_\text{i},\text{c})} \cdot \frac{dc(t,\text{c})}{D} \cdot \log_2 \frac{C+1}{cc(t)+1} \qquad (4)$$

Then we sort the candidate topic words by CTF-CDF-ICF, and select top-k words; in our work, we isolate top $\left\lfloor \frac{D}{2} + 1 \right\rfloor$ words as topic words. This is because the counts of documents in each cluster are different, so we use fewer words to describe the cluster with little documents and more words to describe the cluster with many documents.

4. Generating Topic Trees

4.1 Basic Idea

To build topic hierarchy of terms, Sanderson and Croft's idea [4] can be commonly adopted. Their idea is that for two topical terms t_i, t_j, if $Pr(t_i|t_j) \geq 0.8$ and $Pr(t_i|t_j) > Pr(t_j|t_i)$, then t_i is said to subsume t_j. Here, $Pr(t_i|t_j)$ is the probability that t_i occurs in the document set in which t_j occurs [5]. That is, t_i is a general topic word relatively compared to t_j. In a topic tree, t_i can be t_j's parent and t_j can be t_i's child.

4.2 Modification to the Basic Idea

As seen in Section 4.1, the basic idea for building topic trees is simple yet effective to discover topics that have subsumption relations. However, the idea ignores the topic words' descriptive power. One of our empirical results has showed that topics which are more powerful to describe a cluster are more likely to be the parent in subsumption relations. Therefore, we argue that another parameter for descriptive power should be used when constructing topic words hierarchy.

Thus, our idea is as follows: for two topic words t_i, t_j, if $Pr(t_i|t_j) \geq 0.7$ and $Pr(t_i|t_j) \cdot f(w_i) > Pr(t_j|t_i) \cdot f(w_j)$, then t_i is said to subsume t_j.

$f(w_i)$ is the descriptive power which is evaluated with the CTF-CDF-ICF measure. It is discussed in detail in Section 4.3. In our work, $Pr(t_i|t_j) \cdot f(w_i)$ is the probability of t_i subsumes t_j, not depending only upon $Pr(t_i|t_j)$. In addition, we have modified the first condition of Sanderson and Croft as follows: $Pr(t_i|t_j) \geq 0.7$, not 0.8. This is because the conventional lower bound is too strict in current on-line documents, and actually our new lower bound (i.e., 0.7) have showed the best result in our experiment.

4.3 Normalization of CTF-CDF-ICF

In using CTF-CDF-ICF for building topic trees, we need to consider that the range of CTF-CDF-ICF values are differed from each other cluster, the CTF-CDF-ICF measure should be normalized. Let *W* be the selected topic words' CTF-CDF-ICFs in a cluster: $W = \{w_1, w_2, \ldots, w_n\}$, where w_1 is the largest CTF-CDF-ICF in *W* and w_n is the smallest CTF-CDF-ICF in *W*. We map every CTF-CDF-ICF value in *W* from 1 to (1+d) with the following function:

$$f(w) = \frac{w - w_n}{w_1 - w_n} \cdot d + 1 \qquad (5)$$

where *d* should be a positive number (which is set to 0.5 in our work). Finally, we use the above function to evaluate topic words' descriptive power when determining topic words hierarchy.

5. Experimental Results

5.1 Empirical Setup

To evaluate our proposed method for building topic trees, we have prepared two kinds of datasets, which are collected from Reuters-21578 (*http://www.daviddlewis.com/resources/testcollections/reuters21578/*) and Google news U.S. edition (*http://news.google.com*), respectively. The first dataset, Dataset-1, contains 1000 documents selected randomly in Reuters-21578, and the second one, Dataset-2, contains 1000 documents about business, elections, technology, entertainment, sports, science, and health from August 12, 2012 to September 24, 2012.

We have constructed a topic detection system that can read documents automatically and then show the topic trees. The system is mainly written in JAVA, and the term-processing modules such as stop word removal and stemming is done by R- tm package [6].

And, to evaluate the topic trees generated, we intend to check each pair of subsumption relation in the trees whether it is correct or not. The accuracy is calculated as follows:

$$Accuracy = \frac{\text{the correct number of subsumption relations}}{\text{the total number of subsumption relations}}$$

5.2 Performance Evaluation

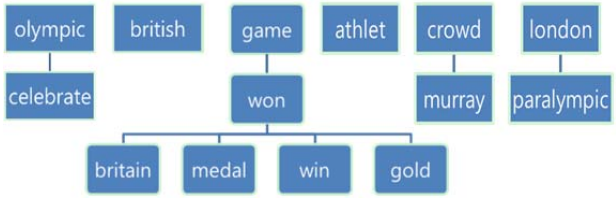
Table 1 shows the accuracy of subsumption relations in topic trees generated by the baseline and the proposed method. The accuracy of Dataset-1, Reuters-21578 dataset, has increased by 17.8%, while the accuracy of Dataset-2, Google news dataset, has increased by 16.1%. Proposed method not only has increased the accuracy of subsumption relations, but also generated topic trees more logical (e.g. Figure 2 and Figure 3).

<Table 1> Accuracy of Subsumption Relations in Topic Trees Generated

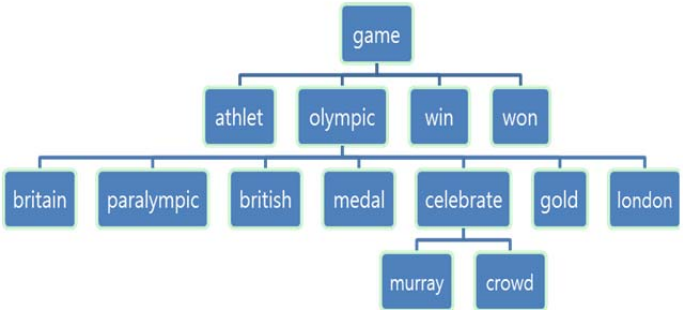
Datasets	Accuracy	
	Baseline	Proposed
Dataset-1	0.743	0.875
Dataset-2	0.769	0.893

Figure 2 is a part of topic trees generated by base line system and Figure 3 is a part of topic trees generated by proposed system. As shown in these figures, we can grasp the main content in both of the figures, but topic trees generated by proposed system are more precise and distinct than base line. Figure 2 shows the topic words in six topic trees and have some incorrect shusumption relations such as relation between "won" and "britain". Figure 3 shows the topic words in one topic tree which is easier to grasp the relations between topics and catch the main topic.

The experimental results suggest that show the topic detection result with topic trees works fairly well using the technique we proposed.



(Figure 2) Topic Trees Generated by Baseline System



(Figure 3) Topic Tree Generated by Proposed System

6. Conclusion

This paper presented a novel technique of discovering significant topic trees for incoming on-line documents. We proposed a novel approach to find topic word subsumption relations for constructing the topic trees. Comparing with basic idea, we consider descriptive power when constructing topic words hierarchy. To get a better result, we used a unique weighting method to get the descriptive power of each word and extracting topic words. To analyze the on-line document, we modified the conventional incremental clustering method into incremental k-medoids algorithm.

In the future, we will try to add up 'noun phrase chunking' to support the separation of phrase, such as "Olympic Games" in Figure 2 and Figure 3. And we will also try to utilize WordNet to resolve the problem of the appearance of "win" and "won", "Britain" and "British" in Figure 2 and Figure 3. Additionally, we will continue our study on incremental clustering to generate more effective clustering results for topic detection.

7. Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number: NRF-2010-0025212).

References

[1] F. Walls, H. jin, S. Sista, and R. Schwartz, "Topic Detection in Broadcast News", Proceedings of the DARPA Broadcast News Workshop, 1999, pp. 193-198.

[2] L. Kaufman, and P.J. Rousseeuw, "Clustering by means of Medoids", Statistical Data Analysis Based on the L1–Norm and Related Methods, 1987, pp. 405–416.

[3] C.D. Manning, P. Raghavan, and H. Schutze, "Introduction to Information Retrieval", Cambridge Univ Press, 2008, pp. 117-120.

[4] M. Sanderson, and B. Croft, "Deriving concept hierarchies from text", Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, pp. 206 – 213.

[5] H.J. Kim, and S.G. Lee, "Building topic hierarchy based on fuzzy relations", Neurocomputing, Volume 51, 2003, pp. 481–486.

[6] I. Feinerer, K. Hornik, and D. Meyer, "Text Mining Infrastructure in R", Journal of Statistical Software, Volume 25, 2008.