Research on Hot Topics Discovery Based on Short Texts of Online Reviews

Zhenyan Liu¹*, Yueshi Qiu¹, Zhe Zhang¹, Limin Mao² and Xiaohan Zheng¹

¹ Beijing Key Laboratory of Software Security Engineering Technology, School of computer science & Technology, Beijing Institute of Technology, Beijing 100081, China
² Academy of Intelligent Collaborative Cloud System, China Aerospace Science and Industry Corporation Limited, Beijing 100048, China

Abstract. The two crucial issues for hot topics discovery based on online reviews are that the sparsity of short text features and the “long tail” phenomenon of hot topics. This paper focuses on these two key issues, and proposes an improved similarity calculation method to calculate the similarity of short texts, and a novel clustering algorithm based on the time factor and dynamic adjustment of comparison times to automatically discard a large number of outliers. Moreover, the validity and advancement of the new method are presented by comparative experiments using real data sets.

1. Introduction

With the rapid development of Internet technology, communication technology and various software and hardware technologies, the popularity of the Internet has increasingly high. Especially in the Web2.0 model, netizens have evolved from “simple reading” to “writing” and further “co-construction”, i.e., from passively receiving Internet information to actively creating Internet information. Through Weibo, BBS (Bulletin Board System), news commentary and blog website, netizens can express their real or personal opinions on various news events or social events in real-time [1]. Further, the interaction between netizens can make various opinions and views clearer quickly, and the active participation of many netizens will make the discussion more extensive and deeper. Therefore, how to extract the hot topics from these real-time, massive and complicated online reviews is a very important and meaningful work.

However, two major challenges must be faced when we extract hot topics from online reviews. One big challenge is that online reviews are mostly short texts. But each short text contains few effective features, so the traditional text similarity calculation methods suited to long text cannot fully play their role. It is necessary to study the text similarity calculation method suited to short text. Another major challenge is that the topic of network hotspots presents an obvious "long tail" phenomenon [2], that is, the number of online reviews that cannot be used to form a hot topic is very large. Thus, the performance of the traditional clustering algorithms owing to this phenomenon is seriously degraded, and then the effective methods must be studied.

This paper will focus on the similarity calculation for short texts and the "long tail" phenomenon in the network hot topics. The paper is organized as follows. In section 2, we briefly review related work. Section 3 describes an improved short text similarity calculation method. In Section 4, we elaborate a novel clustering algorithm based on the time factor and dynamic adjustment of comparison times. Section 5 presents our comparative experiments of this new method. Finally, we conclude the paper in Section 6.
2. Related Work

The traditional process of text clustering typically relies on the Bag of Words model, that is, the words or phrases in the text are not context-related, and their order does not affect the semantics of the text. Yet this model tends to cause the data sparseness when working on short text.

In order to solve the data sparseness problem of short text, Beil et al. [3] have proposed an algorithm named FTC (Frequent Term-based Clustering), which uses some high-frequency words or phrases to represent clusters, and the repeated “frequent terms set” between clusters can be minimized. So there may be many high-frequency terms in the same topic, but the probability of repetition between different topics is smaller. But the disadvantage of this method is that the clustering effect is closely related to the selected order of the frequent terms.

Yang et al. [4] treated each short text as a composition of characters, numbers and punctuation, and have proposed a similarity measure based on string similarity. This method calculates the similarity directly which skips the feature extraction and representation processing of short text, to a certain extent, and avoids using the sparse feature vectors.

Ni et al. [5] have presented a new clustering strategy named TermCut to cluster short text snippets by finding core terms in the corpus. The collection of short text snippets is modelled as a graph in which each vertex represents a piece of short text snippet and each weighted edge between two vertices measures the relationship between the two vertices. TermCut is then applied to recursively select a core term and bisect the graph such that the short text snippets in one part of the graph contain the term, whereas those snippets in the other part do not.

Yin et al. [6] have proposed a collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model for short text clustering (abbr. to GSDMM). GSDMM can infer the number of clusters automatically with a good balance between the completeness and homogeneity of the clustering results, and be converged fastly. GSDMM can also cope with the sparse and high-dimensional problem of short text, and obtain the representative words of each cluster.

Xu et al. [7] have proposed a flexible Self-Taught Convolutional neural network framework for short text clustering, which can incorporate more useful semantic features and learn non-biased deep text representation in an unsupervised manner. In this framework, the original raw text features are firstly embedded into compact binary codes by using one existing unsupervised dimensionality reduction methods. Then, word embeddings are explored and fed into convolutional neural networks to learn deep feature representations.

Summarizing the above related research work, we can find that research on short text clustering emerges endlessly. Many studies have applied different strategies to explore how to deal with short texts. But, there exist few methods to pay special attention to the large number of outliers hidden in massive short texts. However, clustering the short text is seriously impacted by these outliers, in other words, the outliers can result in low efficiency, and poor effect of clustering. Therefore, according to the characteristics of online reviews, it is necessary not only to focus on the method for short text similarity calculation, but also to pay attention to how to deal with the large number of outliers hidden in the data. These two parts are just the focus of this study, which will be elaborated below.

3. Similarity Calculation Method for Short Texts

The clustering algorithm is usually based on the vector space model (abbr. to VSM) [8], that is, the feature space is firstly determined by a feature selection method, such as the document frequency (abbr. to DF), and then the weight of feature term is calculated by TFIDF(Term Frequency Inverse Document Frequency) [9] to represent texts. Next these represented texts are clustered by the clustering algorithm, in which the key similarity calculation method is mostly Euclidean distance or cosine similarity.

However, this scheme is mainly suitable for long text, it will be powerless when it is used for short text. Because the length of short text is short, for example, a piece of weibo with only little number of terms. Generally, a term appears no more than twice in a short text. Therefore, for each term, TF value is almost the same, and the weight of a term is more affected by IDF. The main purpose of IDF is to reduce the weight of generic terms. It simply thinks that the more important terms appear in fewer documents, and terms appeared in the more documents are relatively unimportant. In hot topics
discovery, TF has lower distinguishing ability, so the weight of a term is determined almost exclusively by IDF which cannot accurately represent the distribution of feature terms, so the TFIDF effect is not ideal. In addition, A more important reason is that short texts have few effective features, which leads directly to the high sparsity of text vectors. Thus the text similarity computed by Euclidean distance or cosine similarity will be unsatisfactory.

Based on the above analysis, this paper employs an improved Jaccard similarity coefficient to evaluate the similarity of short texts, which will be implied in the novel clustering in next section. The improved Jaccard similarity coefficient adds the length factor of the short text to the traditional Jaccard similarity coefficient [10]. Because the relative lengths of short texts are quite different, a term plays a much greater role in shorter text than it does in longer text. This coefficient is described in detail as follows.

For the two short texts $T_1$ and $T_2$, they are expressed as follows:

$$T_1 = t_{11} \ t_{12} \ ... \ t_{1m}$$

$$T_2 = t_{21} \ t_{22} \ ... \ t_{2n}$$

Where $t_{11} \ t_{12} \ t_{13} \ ... \ t_{1m}$ is a sequence of terms from $T_1$ which is preprocessed by word segmentation, removing stop words, etc. $t_{21} \ t_{22} \ ... \ t_{2n}$ is from $T_2$.

$T_1$ and $T_2$ have m and n terms, respectively. And the value of each term is assigned 0 or 1 based on a selected vocabulary (size=N). Define the following three statistics:

$$A_{01} = \sum_i C(T_{1i} = 0 \land T_{2i} = 1)$$

$$A_{10} = \sum_i C(T_{1i} = 1 \land T_{2i} = 0)$$

$$A_{11} = \sum_i C(T_{1i} = T_{2i} = 1)$$

Where $C(x)$ is a truth function, if x is true then $C(x) = 1$, else $C(x) = 0$, and $i = 1,2,...,N$. Next, we define the formula of the similarity of $T_1$ and $T_2$ as follows.

$$J(T_1, T_2) = \frac{A_{11}}{A_{01} + A_{10} + A_{11}}$$

Where $T_1$ and $T_2$ are the length of $T_1$ and $T_2$, respectively.

Compared with the traditional similarity computation method, the improved Jaccard similarity coefficient fully considers the characteristics of short text, which is especially suitable for short text data with excessive sparsity.

4. AdaKmeans Clustering Algorithm

There is a lot of noise (i.e. outliers) for clustering in production data. When the number of outliers increases to a certain magnitude, almost all clustering algorithms cannot guarantee the clustering effect. For data sets like web short text, the number of outliers far exceeds the number of valuable big clusters. For the massive corpora, an outlier will not defined a point in the traditional sense, but a very small cluster in a broad sense. We randomly select 10,000 pieces of data for analysis and statistics. Through multiple statistical experiments, we find that there exist "long tail" phenomenon in the hot topics of online reviews obviously. As shown in Figure 1.
Figure 1. The "long tail" phenomenon of hot topics.

The actual application should focus on the left part of Figure 1. Each single cluster in this area contains a large number of short texts. These clusters are just the hot topics to be identified. Each single cluster in the "long tail" area presented on the right part of Figure 1 contains a small number of short texts, which are not hot topics and need no attention. However, the number of such small clusters is especially large, which are just the noise that affect the clustering effect.

In order to discard the small clusters in the "long tail" region and focus on the valuable clusters while clustering, this paper proposes an improved Kmeans algorithm based on the characteristics of network texts. The key idea of the improved algorithm is to introduce the time factor and dynamically adjust comparison times. We named this algorithm Adaptive Kmeans (abbr. to AdaKmeans).

Firstly, we find that the appearance of short texts related to a topic in practical applications is relatively compact in time. That is, if some short texts around a topic appear, the probability that the short texts appearing in the subsequence adjacent time belongs to this topic is greater than that of the short texts appearing in a long time interval. The actual explanation is that an even may be gradually heated up by people, and the number of related short texts will rise sharply in a short period of time. Over time, especially with the emergence of new hot events, the number of short texts about this topic will gradually decrease until it fades out of people's sight.

Secondly, comparison times between the new sample and the existing category will be adjusted dynamically. The main idea of this dynamic adjustment is, for a new sample that needs to be processed, to compare K' (K'<K) categories selected dynamically based on the time factor of the new sample, and then the new sample will be clustered into the most similar category and the similarity between them exceeds a preset threshold. If the similarity of the new sample to each category is less than the preset threshold, it is regarded as a new category.

The key improvements of the AdaKmeans algorithm are as follows:

1) The initial cluster number Selection: According to a given K' value, in chronological order, for all the data in the first time window, we randomly select K' objects as the initial cluster center, and then completely cluster all the data into K' clusters based on Kmeans algorithm, and let K=K'.

2) Clustering: For the data in the next time window, the Kmeans algorithm is also executed to obtain the K' cluster. The cluster centers of the K' clusters in the current time window are viewed as new samples which are compared with the existing K clusters. And the two clusters whose similarities are the highest as well as exceed the preset threshold will be merged. If the similarity between a new cluster and an existing K cluster does not exceed a preset threshold, it will be added as a new cluster, and let K=K+1.

3) The optimization of the previous step: The new cluster is not compared with all the existing K clusters, but the o(o<K) clusters with the most recent time (The reason is that hot discussion has a time-dependent nature). The o value is selected according to the aggregation degree of time. If the new cluster is merged into the current K clusters, the time aggregation gain of each new cluster is calculated, and then the top o clusters are selected. The reason is that the closer in time the samples are, the more likely they are to belong to the same topic. If the time of the two clusters is quite
different, the probability they belonging to the same cluster is very low, and there is no need to compare the similarity between the clusters.

4) The further optimization of the previous step: As the clustering process proceeds, the clusters will be more and more, especially some hotspots are no longer hot. In order to dynamically adjust the number of clusters K, and make the current clustering processing task more lightweight, we set a flag for each cluster. If a cluster is not updated within several time windows, the cluster is no longer a "hotspot". So we will save this cluster which will no longer participate in the next round of clustering, and let K=K-1.

5. Experiment

In order to validate the effectiveness of AdaKmeans algorithm, we use 10000 online reviews in a historical period as experimental data to compare the basic Kmeans algorithm with the AdaKmeans algorithm. The experimental results are shown in Table 1, in which the number of categories of the basic Kmeans algorithm is initially artificially set and doesn’t change anymore, and the number of categories of the AdaKmeans algorithm is finally obtained by the cluster algorithm. (The number of initial category is also set to 40).

The basic Kmeans algorithm does not handle outliers, so there are no outliers. The improved AdaKmeans algorithm’s outliers are some single texts or some smaller clusters. In order to test the clustering effect, all the experiment data are labeled artificially. The basic Kmeans algorithm is affected by outliers, so its accuracy rate is calculated on the basis of removing outliers.

| Number of categories | Outliers | Accuracy rate | Runtime |
|----------------------|----------|---------------|---------|
| Kmeans               | 40       | ---           | 65%     | 1195s   |
| AdaKmeans            | 56       | 5036          | 88%     | 588s    |

As can be seen from Table 1, it takes about 20 minutes to cluster 10,000 data using the basic Kmeans algorithm. And because of the high noise, even if the noise data are not taken into account in the result statistics, the accuracy rate is very low. In fact, the number of categories may be more than 40, or less than 40. This kind of manual specifying the number of clusters based on experience has poor clustering effect.

For the improved AdaKmeans algorithm, when the initial number of categories is also set to 40, the final number of categories is 56, that is, in the corpus, there are 56 categories identified by the program as the larger categories. It’s clearly indicated that, when using the basic Kmeans, some samples are misclassified. Compared with the basic Kmeans, the accuracy and running time of the AdaKmeans are greatly improved.

From the above results, we can see that the "long tail" phenomenon implied in the actual online reviews is indeed a difficult problem for the basic clustering algorithm. And the improved algorithm based on the basic clustering algorithm can solve this problem well, which can significantly improve the efficiency of the program and the clustering effect.

6. Conclusions

Discovering hot topics based on online short texts is a research hotspot in the field of social public opinion situational awareness. According to the remarkable characteristic of short text with few effective features and the "long tail" phenomenon implied in online reviews, this paper proposes an improved Jaccard similarity coefficient to measure the similarity of short texts, and introduces it into the improved AdaKmeans clustering algorithm.

In addition to introducing a new short text similarity method, the improved AdaKmeans also solves the problem of how to automatically discard a large number of outliers in the clustering process based
on the time factor and dynamic adjustment comparison times. The solution of this problem improves the performance of clustering algorithm and the effect of hot topics discovery greatly.

However, in the process of validating the algorithm, we find that it is still difficult to set two key parameters: similarity threshold parameter and time aggregation parameter, which will directly affect the final clustering effect. Therefore, one of the next priorities is to further study the optimization strategies of these two parameters. In addition, the amount of experimental data is relatively small at present, so how to apply big data technology to solve the online clustering problem of big data will also be the next research focus.

7. References
[1] Lan Y, Liu B, Zhang P, Xia Y, Li H. The Internet Public Opinion Hot-degree Dynamic Prediction Model Oriented to Big Data [J]. Journal of Intelligence. pp. 36(6):105-110, 2017.
[2] Oestreichersinger G, Sundararajan A. Recommendation Networks and the Long Tail of Electronic Commerce [M]. Society for Information Management and the Management Information Systems Research Center, 2009.
[3] F.Beil, M.Ester. Frequent Term-Based Text Clustering [J]. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 436-442, 2002.
[4] Yang Z, Duan L, Lai Y. Online Public Opinion Hotspot Detection and Analysis based on Short Text Clustering Using String Distance [J]. Journal of Beijing University of Technology, 36(05): 669-673, 2010.
[5] Ni X, Quan X, Lu Z , et al. Short text clustering by finding core terms[J]. Knowledge and Information Systems, 27(3):345-365, 2011.
[6] Yin J, Wang J. A dirichlet multinomial mixture model-based approach for short text clustering[C]. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 233-242, 2014.
[7] Xu J, Xu B, Wang P, et al. Self-Taught Convolutional Neural Networks for Short Text Clustering [J]. Neural Networks, 88:22-31, 2017.
[8] G.Salton, A.Wang, C.S. Yang. A Vector Space Model for Automatic Indexing [J]. Communication of the ACM, 18(11): 613-620, 1975.
[9] Jing L P, Huang H K, Shi H B. Improved Feature Selection Approach TFIDF in Text Mining [M]. Improved feature selection approach TFIDF in text mining, 2002.
[10] Wu C, Wang B. Extracting Topics Based on Word2Vec and Improved Jaccard Similarity Coefficient[C]. Proceedings of the IEEE Second International Conference on Data Science in Cyberspace. pp. 389-397, 2017.