ISSN 0146-4116, Automatic Control and Computer Sciences, 2016, Vol. 50, No. 4, pp. 271–277. © Allerton Press, Inc., 2016.

Micro-Blog Topic Detection Method Based on BTM Topic Model and K-Means Clustering Algorithm[1]

Weijiang Li[a, *], Yanming Feng[a], Dongjun Li[b], and Zhengtao Yu[a]

[a] Department of Information Engineering and Automation,
Kunming University of Science and Technology, Kunming, 650500, China

[b] R&D Department Jinan Qingqi Peugeot Motorcycle Co. Ltd., Soochow University, Jinan, Shandong, 250104, China

*e-mail: hrbrichard@126.com

Received December 11, 2015; in final form, May 23, 2016

Abstract—The development of micro-blog, generating large-scale short texts, provides people with convenient communication. In the meantime, discovering topics from short texts genuinely becomes an intractable problem. It was hard for traditional topic model-to-model short texts, such as probabilistic latent semantic analysis (PLSA) and Latent Dirichlet Allocation (LDA). They suffered from the severe data sparsity when disposed short texts. Moreover, K-means clustering algorithm can make topics discriminative when datasets is intensive and the difference among topic documents is distinct. In this paper, BTM topic model is employed to process short texts – micro-blog data for alleviating the problem of sparsity. At the same time, we integrating K-means clustering algorithm into BTM (Biterm Topic Model) for topics discovery further. The results of experiments on Sina micro-blog short text collections demonstrate that our method can discover topics effectively.

Keywords: short text, topic model, topic discovery, K-means clustering

DOI: 10.3103/S0146411616040040

1. INTRODUCTION

Micro-blog is a social platform where we can share kinds of information and acquire hot topic. Moreover, the characteristic of it is the limit of the word number. When we post the fresh, our texts should be within 140 words. The simple and concise content can be posted in real-time and be spread rapidly, when a lot of short text generate. So discovering hot topics from numbers of short texts can make us know the latest and discussed topics. These topics are necessary for guiding public sentiment and controlling the network. Furthermore, it has influences on topic evolution trend and information security. Thus, studying topic discovery from micro-blog can bring us theoretical research value and social value [1].

Topic discovery experiences preprocessing, modeling, similar calculation and so on. Specially, modeling plays a vital role. Originally, people proposed vector space model (VSM) and statistic language model. However, they didn't take semantic into consideration, so that they couldn't capture topics well. Then, PLSA [2] added statistics to improve LSA, and LDA [3] was proposed by D. Blei. They resort document-level word co-occurrence to capture topics. Moreover, LDA gained great effect on normal texts in topic detection. But, the length of micro-blog is short and contains less content, so that less document-level word co-occurrence appears [4]. What's more, the randomness of express in micro-blog generated confined context and less related topic words. The two aspects make conventional topic models, such as PLSA and LDA, suffer from the greatest challenge of discovering topics from micro-blog. The challenge is data sparsity problem.

We do final step according some strategy in topic discovery. For example, Single-Pass clustering algorithm [5] is a kind of method used in topic discovery currently. It uses incremental clustering algorithm to deals with news report. The similarities between the feature vector of news reports and centers of historical topics have an important impact on the performance of topic detection, and hierarchical clustering [6] is fit for similarity calculation and distance vector. Moreover, K-means [7] is a widely used clustering algorithm. The calculation speed of K-means is fast, and its clustering result looks desired. All the clustering algorithms have their merits. However, when data set is concentrated and different topics have distinct meaning, the result of K-means clustering is better.

[1] The article is published in the original.

For the topic model, we know BTM proposed by Yan Xiaohui, which can discover more prominent and coherent topics from short text [4]. Moreover, combining the K-means clustering algorithm can get satisfied result. Inspired from Extracting News Topics from Micro-blogs based on Hidden Topics Analysis and Text Clustering [8] in this paper, we proposed the idea that combining BTM Topic Model with K-means clustering algorithm. We made full use of BTM to acquire topics from short texts and K-means clustering algorithm to help us cluster topics better. Experiment results show our method works well.

2. RELATED WORK

2.1. The Method of Alleviating Data Sparsity

According to previous research, it is hard to implement text mining on micro-blog. Some traditional mining technology generated inevitable problems when processing micro-blog data. Namely, the severe data sparsity made them less effective on short texts.

LDA is a kind of general, stable and unsupervised topic model. People generally conducted more improvement based on LDA to deal with data sparsity in short text [9]. On one hand, they focused on external information to make up the representation of short texts. For example, Xiaoqing Qi and Xiaojun Jing [10] suggested an improvement of LDA applying in micro-blog – ULLDA. Taking the relation between user information and text relevance, they expanded LDA model by means of user and link. On the other hand, some researchers expanded traditional topic model to alleviate limited texts. Such as, Hao Xie and Hong Jiang [9] proposed RT-LDA micro-blog generative model based on LDA to resolve the limit of word. Moreover, they adopted Gibbs to mining the topic of every piece of post. Daniel Ramage, Susan Dumail, Dan Liebling [11] presented a scalable implementation of a partially supervised learning model (Labeled LDA) that maps the content of the micro-blog feed into dimensions. Then to made use of this model to characterize users and tweets.

However, to gather external information costs a certain amount of work, and this kind of model relied on external data. For example, ULLDA may be helpful in summarizing the topic that users concerned, but it relied on the information of users. Moreover, expanded models have some limitations. We know RT-LDA made an assumption that every piece of micro-blog only contains one topic. Obviously, this kind of assumption ignored else existing topics.

2.2. Topic Discovery in Micro-blog

Early studies primarily paid attention to exploit the clustering algorithm to discover topic from short text. Ting Huang, Dunlu Peng and Lidong Cao [12] made use of self-adaptive *k* clustering to discovering communities in micro-blog data. They developed CLARANS algorithm and combined a self-adaptive *k* strategy to enhance this idea. Shengping Sun [13] proposed SP&HA clustering algorithm based on vector space model for detection. He expressed text by vector space mode and calculated it by SP&HA clustering algorithm. In fact, if these methods are combined with topic model, they will get more effective results.

Presently, people prefer to use topic model and text clustering to discover topics and track topics. Zutao Xiong [8] proposed clustering algorithm research in micro-blog short text based on sparse feature. He employed LDA to express texts. Then he combined K-means and hierarchical clustering algorithm to cluster micro-blog data. Wenli Mi and Yuexin Sun [14] developed micro-blog hot topics discovery method based on probabilistic topic model. This paper modeled the microblog topic by pLSA (probabilistic Latent Semantic Analysis), and generated topic by K-means. Then to discover current hot topic through topic degree and sequence. Rong Lu, Liang Xiang, Mingrong Liu and Qing Yang [7] proposed to use potential topic model to mining topic information from short texts. What's more, K-means and hierarchical clustering algorithm were utilized to clustering news topics.

3. BTM TOPIC MODEL AND K-means ALGORITHM

3.1. BTM Topic Model

A bi-term denotes an unordered word-pair co-occurring in short texts [4]. As we all know, every piece of micro-blog (document) contains little contents, each document is taken as an independent text unit. Any two distinct words can make up a bi-term. For example, "the stampede in shanghai bund." We can extract "stampede shanghai," "stampede bund," "shanghai bund." The three bi-terms are what we need to extract; and in all the documents, we extracted bi-terms like this to compose the training data of BTM.

Different from LDA, BTM models the generation of bi-terms with latent topic structure, rather than modeling the generation of documents with latent topic structure. Suppose α and β are the Dirichlet priors. The generative process of the corpus in BTM is represented as Fig. 1.

To get a topic assignment $z \sim \mathrm{Multi}(\theta)$ and two words: w_i, $w_1 \sim \mathrm{Multi}(\phi_z)$ for each bi-term b in the bi-term set B. Firstly, we draw a topic – specific word distribution $\phi_z \sim \mathrm{Dir}(\beta)$ for each topic z. Then, to describe a topic distribution $\theta \sim \mathrm{Dir}(\alpha)$ for the whole collection. Following the above procedure, we can write the joint probability of a bi-term $b = (w_i, w_j)$ like this:

$$P(b) = \sum_z P(z)P(w_i|z)P(w_j|z) = \sum_z \theta_z \phi_{i|z} \phi_{j|z}. \quad (1)$$

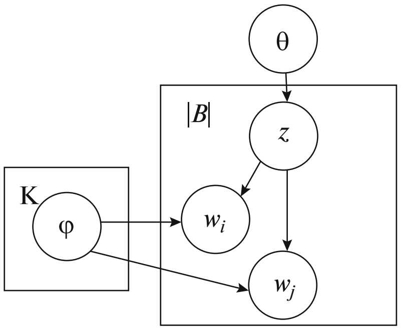

Fig. 1. Graphical model of BTM.

Thus, the probability of the whole corpus $P(B)$ is as follows:

$$P(B) = \prod_{(i,j)} \sum_z \theta_z \phi_{i|z} \phi_{j|z}. \quad (2)$$

From this, we can know BTM directly model the word co-occurrence pattern, which can reveal the topics effectively, and enhance the learning of topics. We cannot get the topic proportions of documents, when we learning the topics. To infer the topics, the expectation of the topic proportions of bi-terms is adopted:

$$P(z|d) = \sum_b P(z|b)P(b|d), \quad (3)$$

and

$$P(b|d) = \frac{n_d(b)}{\sum_b n_d(b)}. \quad (4)$$

Here we simple take the empirical distribution of bi-terms in the document as the estimation, where $n_d(b)$ is the frequency of the bi-term b in document d; $\sum_b n_d(b)$ is sum of bi-terms' frequency in document d; $P(b|d)$ is nearly a uniform distribution over all bi-terms in document d, which is the percent of bi-term b in all bi-terms.

3.2. Inference by Gibbs Sampling

As we learn from LDA, similarly, we can know ϕ and θ are connotative variable in BTM topic model. Accordingly, they need to be estimated on the basis of words observed from documents. Generally, exact inference and approximate inference are used to deal with the inference of probability model. With the difficulty of exact inference, we use approximate inference to learn the potential variable.

In views of universality and comprehension, we estimate ϕ and θ by means of Gibbs Sampling. And the main idea of it is to change one of the variables, with the value of others invariant. Iterating the course, the appearance of convergence can generate parameters' output.

With Markov chain, we can get the following conditional probability distribution for each bi-term:

$$P(z|z_{-b}, B, \alpha, \beta) \propto (C_{bz}^{BT} + \alpha) \frac{(C_{iz}^{WT} + \beta)(C_{jz}^{WT} + \beta)}{\left(\sum_w C_{wz}^{WT} + M\beta\right)^2}, \quad (5)$$

where $P(z|z_{-b}, B, \alpha, \beta)$ is conditional probability distribution for each bi-term, and z_{-b} represents all topic assignments for bi-terms, not including the b-th bi-term; B is the set of bi-terms, and C_{bz}^{BT} is the number of times of bi-term b assigned to the topic z, not including the current instance; C_{wz}^{WT} is the word w assigned to topic z; C_{iz}^{WT} is the word i assigned to topic z, and the same to C_{jz}^{WT}.

Finally, after calculating the topic assignments of bi-term, we can obtain the estimation of ϕ and θ as follows:

$$\phi_{w|z} = \frac{C_{wz}^{WT} + \beta}{\sum_{w} C_{wz}^{WT} + M\beta}, \tag{6}$$

and

$$\theta_z = \frac{C_{bz}^{BT} + \alpha}{n_b + T\alpha}, \tag{7}$$

where $\phi_{w|z}$ is topic-word distribution; θ_z is topic distribution, and n_b is the totality of bi-terms; α and β are the Dirichlet priors; C_{wz}^{WT} is the word w assigned to topic z, and T is the number of topics.

3.3. Clustering Algorithm

After the mining of potential topics, we adopt K-means clustering to cluster the hot topics, liking [15]. K-means is a clustering algorithm based on division. K-means clustering is simple and widely applicable. It needs the number of cluster before implement, and then data will be divided to several clusters. K-means chooses several documents as initial cluster center at first. Moreover, it will calculate average value and assign documents to new cluster. Finally, update center until convergence.

The basic idea of K-means is as follows:

Algorithm 1: K-means algorithm based on BTM

Input: the number of cluster J, topic sets obtained by BTM topic model B

Output: cluster set D

Steps:

(1) choose J data point as cluster center from D;

(2) distribute every data point to a nearest center x;

(3) calculate mean value of topics to update the center of cluster

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}; \tag{8}$$

(4) until the topics reach convergence

$$\lim \sum (x_i - x)^2 = 0. \tag{9}$$

When cluster is compact, and the difference between clusters is obvious, K-means can obtain better cluster result [16]. Then because BTM model can discover different topics effectively, we can get intensive data result, and the difference between topic documents is distinct. Based on this, we apply K-means algorithm to gain topic cluster. Thus, we can guarantee the robustness of K-means algorithm.

4. EXPERIMENTS

In this section, we conduct experiments on Sina Micro-blog collections to demonstrate the effectiveness of BTM and K-means. We will compare the result between BTM and HC and BTM and K-means. The result will explain that K-means performs better than HC, when the difference between topic documents is distinct.

4.1. Datasets

We collect datasets from Sina micro-blog randomly, and we get 10 kinds of topic. ‘School season’, ‘Anti Japanese military parade’, ‘Reading’, ‘Tianjin explosion’, ‘Shanghai trample’, ‘Finance’, ‘Car’, ‘Computer’, ‘Sports’, ‘Recruitment’. Each of them contains about 100 pieces of micro-blog.

Table 1. Top 5 words drawn from top 3 topics

Top 5	Topic 1	Topic 2	Topic 3
1	Trample	Taijin	War
2	Tread	Bomb	Parade
3	Event	Tangku	Anniversary
4	Shanghai	Accident	70
5	Bund	Attention	Victory

Table 2. Non-top words

Non-Top 3	Topic 1	Topic 2	Topic 3
1	List	Discuss	Chinese
2	Announce	Event	Encounter
3	Die	Rare	Upsurge

To improve the effectiveness of this content, we use NLPIR (ICTCLAS2013) to process the raw blog as follows (1) segment words (2) remove stop words (3) wipe off individual character (4) remove the micro-blog with @user (5) delete the duplicate.

4.2. The Result of BTM

After processing raw micro-blog, we use BTM topic model to obtain topics of micro-blog. We set *k* as 10, we only take top 3 topics to compare their coherence. From Table 1, we can know they belong to the topic ‘Shanghai trample’, ‘Tianjin explosion’, ‘Anti Japanese military parade’.

On the other hand, we list the non-top words of these topics in Table 2. The words of first topic displays announce to death list, the second explain that the event of Tianjin Tangku has aroused people’s discussion. Moreover, ’Chinese’, ’encounter’ and ’upsurge’ state that Chinese people keep rising to Anti Japanese military parade. Then, what we want to express is non-top words keep consistency to top words. BTM topic model can discover potential topic effectively. Meanwhile, BTM can improve the problem of sparse data, which LDA suffers from.

4.3. Evaluation on Clustering Algorithm

Based on BTM, we use clustering algorithm to cluster discovered topic. In this way, we can avoid repeated and crossed topic. On the other hand, we can guarantee topics be comprehensive.

After BTM, Weka helps to realized cluster. We make use of ‘TextDirectoryLoader’ and ‘String To Word Vector’ to process topic texts. Then we take hierarchical clustering as our baseline to make evaluation to K-means clustering. We know precision and recall can reflect the accuracy, degree of check. If we want the related topic, we may pursue precision. On the contrary, we request recall when we expect discovered topic be more. In this paper, we adopt *F*(*F*-measure) to measure the result of cluster, which can combine precision with recall.

The expression of *P*(precision), *R*(recall), *F*(*F*-measure) is as follows:

$$P(i,j) = \frac{\text{the number of class } i \text{ in cluster } j}{\text{the number of documents in cluster } j}, \tag{10}$$

$$R(i,j) = \frac{\text{the number of class } i \text{ in cluster } j}{\text{the whole number of documents with class } i}, \tag{11}$$

$$F(i) = \frac{2PR}{P+R}, \tag{12}$$

Table 3. The F of hierarchical clustering and K-means with different numbers of cluster when the number of topics is 7

Cluster	HC–F	K-means–F
5	0.655	0.649
6	0.596	0.798
7	0.770	0.774
8	0.768	0.845

Table 4. The F of hierarchical clustering and K-means with different numbers of cluster when the number of topics is 10

Cluster	HC–F	K-means–F
4	0.333	0.515
5	0.490	0.575
6	0.576	0.718
7	0.697	0.743
8	0.837	0.849

where i is class, and j denotes the cluster. Moreover, for i, the cluster with high F represents the class i. Finally, we apply weighted average to every class to get the final results of clustering.

$$F = \frac{\sum_i [|i| \times F(i)]}{\sum_i |i|}. \tag{13}$$

We compare K-means with hierarchical clustering, when the number of topics is 7 and 10. Then test F-measure with hierarchical clustering and K-means for different clusters. Table 3 displays F when the number of topics is 7. Furthermore, we compare the two clustering for the number of cluster being 5, 6, 7, 8, and Fig. 2 tells us that F-value for K-means is higher than that for hierarchical clustering. When the number of cluster is closes to topic number, the value of F increases.

When we get 10 topics through BTM, we set the number of cluster for 4, 5, 6, 7, 8, 9, 10, 11, 12 to obtain F. Table 4 illustrates the effect of K-means is better than hierarchical clustering, when combining with BTM topic model to discover topics. From Fig. 3, we know F ranges from 0.3–1.0. When clusters conform to real topics number, F is close to 1.0. Because the number of topic is 10, even if we set the number of cluster as 11 and 12, the number of cluster displayed is still 10, and F value keeps being 1.0.

From Fig. 3, we know the F-measure is perfect, but in Fig. 2, the F-measure in case of 7 topics is less than perfect. Through analysis, we consider the ambiguity of topic may affect the F-measure of clustering. The ambiguity makes the topic belong more clusters.

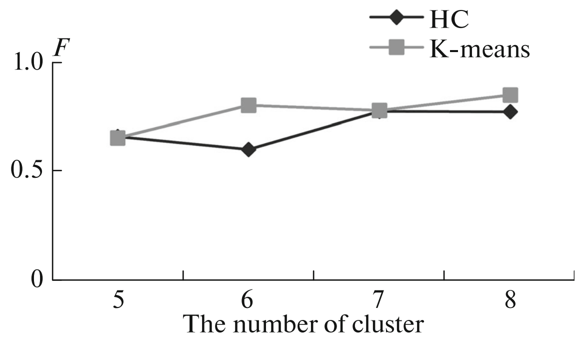

Fig. 2. The effect of cluster numbers for 7 topics.

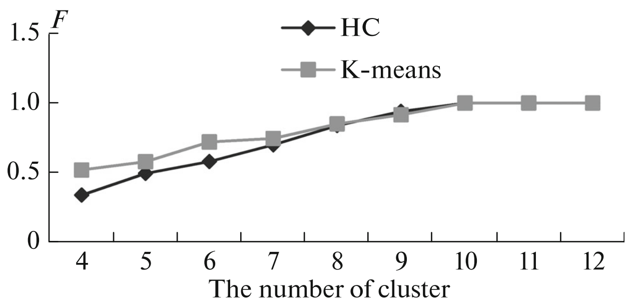

Fig. 3. The effect of cluster numbers for 10 topics.

5. CONCLUSION

Topic discovery is an increasingly significant task, because the effect of topic spread is inestimable. Compared to normal text mining, discover potential topics from micro-blog will suffer the sparsity of word and information content. On the other hand, the selection of clustering algorithm is should be considered and compared. In this paper, we proposed the combination of BTM model with K-means to process sparsity problem and improve the result of clustering. We use our approach on Sina micro-blog, and obtain desired topic with high *F* value.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (61363045); The Key Project of Yunnan Nature Science Foundation (no. 2013FA130); Science and technology innovation talents fund projects of Ministry of Science and Technology (no. 2014HE001).

REFERENCES

1. Huang, S.Q., Yang, Y.T., Li, H.K., and Sun, G.Z., Topic detection from microblog based on text clustering and topic model analysis, *IEEE Asia-Pac. Serv. Comput. Conf.,* 2014, no. 12, pp. 88–92.
2. Hofmann, T., Probabilistic latent semantic indexing, *Proc. SIGIR,* 1999, pp. 50–57.
3. Blei, D., Ng, A., and Jordan, M., Latent Dirichlet allocation, *J. Mach. Learn. Res.,* 2003, vol. 3, pp. 993–1022.
4. Yan, X.H., Guo, J.F., Lan, Y.Y., and Cheng, X.Q., A biterm topic model for short texts, *Int. Conf. World Wide Web,* 2013, no. 5, pp. 1445–1456.
5. Liu, S.B. and Liu, L., Combining parametric and nonparametric topic model to discover microblog event, *IEEE Inf. Sci. Electron. Electr. Eng. (ISEEE),* 2014, vol. 3, pp. 1527–1531.
6. Wang, Y.Y., Wang, L., Qi, J., et al., Improved text clustering algorithm and application in microblogging public opinion analysis, *IEEE Fourth World Congress on Software Engineering,* 2013, pp. 27–31.
7. Lu, R., Xiang, L., Liu, M.R., and Yang, Q., Discovering news topics from micro-blogs based on hidden topics analysis and text clustering, *Pattern Recognit. Artif. Intell.,* 2012, vol. 3, pp. 382–387.
8. Xiong, Z.T., Clustering algorithm research in micro-blog short text based on sparse feature, *Software Guide,* 2014, vol. 13, pp. 133–135.
9. Xie, H. and Jiang, H., Improved LDA model for micro-blog topic mining, *J. East China Nornal Univ. (Nat. Sci.),* 2013.
10. Qi, X.Q. and Jing, X.J., The improvement of LDA applying in micro-blog, *Sci. Pap. Online,* 2012.
11. Ramage, D., Dumail, S.T., and Liebling, D.J., Characterizing micro-blogs with topic model, *4th International AAAI Conference on Weblogs and Social Media (ICWSM),* 2010, pp. 130–137.
12. Huang, T., Peng, D.L., and Cao, L.D., Discovering communities with self-adaptive k clustering in micro-blog data, *IEEE Second International Conference on Cloud and Green Computing,* 2012, pp. 383–390.
13. Sun, S.P., *Research on Chinese Micro-Blog Hot Topic Detection and Tracking,* Beijing Jiaotong University, 2011.
14. Mi, W.L. and Sun, Y.X., Microblog hot topics discovery method based on probabilistic topic model, *Comput. Syst. Appl.,* 2014.
15. Zheng, L., *Research and Application of Topic Detection on Micro-Blog,* Harbin Institute of Technology, 2012.
16. Han, J.W. and Kamber, M., *Data Mining: Concepts and Techniques [M],* 2007, pp. 263–266.