Two-Stage Clustering Hot Event Detection Model for Micro-blog on Spark

Ying Xia(✉) and Hanyu Huang

School of Computer Science and Technology,
Chongqing University of Posts and Telecommunications, Chongqing 400065, China
xiaying@cqupt.edu.cn, hanyuhuang.hhy@gmail.com

Abstract. With the rapid development of micro-blog, it has become one of the main platforms to publish news and express opinions. Micro-blog analyzing for hot event detection is widely concerned by researchers. However, hot event detection is not easy because micro-blog blogs have the characteristics of large scale, short text and irregular grammar. In order to improve the performance of hot event detection, a two-stage clustering hot event detection model for micro-blog is proposed. The model is designed in spark environment and divided into two parts. First, K-Means method is improved by threshold setting and cosine similarity to cluster blogs. Then, the result of blogs clustering is clustered again to detect hot events by LDA (Latent Dirichlet Allocation) model. Sufficient experiments have been carried out in spark environment, it is shown that the proposed model gains higher accuracy and time efficiency for hot event detection.

Keywords: Micro-blog blogs · Hot event detection · Spark · Two-stage cluster · K-Means model · LDA model

1 Introduction

Hot event refers to event with high public discussion and widespread concern. Timely detection of hot event has great significance for society management and public safety maintenance. Micro-blog is an important online communication media, hot event can be considered when a large number of blogs discussing a same topic. To detect hot events, researchers have proposed different models. These models can be divided into two categories, keywords extraction model and topic model.

Keywords extraction model can analyze and extract keywords from blogs. Extracted keywords are used to cluster texts and then detect events. Early research [9] paid more attention to extract keywords. But only extract keywords may cause insufficient semantic information. To solve this problem, researchers combine related features and keywords to detect events. Stilo et al. [12] and Ozdikis et al. [11] used Hashtag to enhance accuracy of event detection. Furthermore, different features are added according to different research objectives.

S. Wen et al. (Eds.): ICA3PP 2019, LNCS 11945, pp. 175–183, 2020.
https://doi.org/10.1007/978-3-030-38961-1_16

Sun et al. [13] combined external knowledge base of related fields to detect events. Yilmaz et al. [17] and Zhong et al. [18] mixed geographical position and keywords to detect location-related events. In addition, in order to improve efficiency of event detection and ensure accuracy, keywords clustering process are focused. Ai et al. [1] proposed a TMHTD model in Spark, it detects events via calculating the similarity of keywords in two-layers structure.

Compared with keywords extraction model, topic model gains higher event detection accuracy but needs sufficient features. LDA model [2] is a representative topic model, it uses word bags to describe events and no need to consider words order in texts. Hao et al. [5] used LDA model to extract topics while identifying abnormal behavior sentences with each topic. Wang et al. [14] visualized topics after extracting topics from LDA model. In addition, some research try to improve accuracy of topic model by expanding feature space [3,4,6,7]. However, complex features are difficultly added due to the limitation of model structure. Some research extended semantic information to further improve accuracy. Yan et al. [16] and Kitajima et al. [8] used advanced semantic information like binary or triple sets instead of word bags for clustering. Xu et al. [15] proposed a TUS-LDA model which used pseudo-texts as the input of LDA model. The pseudo-texts were clustered by different topic types to expand semantic information.

Considered the advantages of the two categories of models, a two-stage clustering hot event detection model is proposed. These two stages are named text-cluster stage and semantic-cluster stage, respectively. In which, K-Means model and LDA model are involved in clustering process according to data characteristics of different stages, K-Means model is optimized by threshold setting and cosine similarity for keywords extraction, and a set of spark jobs is designed for large-scale data processing.

The rest of paper is organized as follows. Section 2 presents terminology definition. Section 3 proposes the two-stage clustering hot event detection model. Section 4 designs the optimized model in Spark environment. Section 5 evaluates accuracy and efficiency of proposed model. Section 6 draws a summary.

2 Terminology Definition

For easily understanding, related terminologies are presented here.

Micro-blog Blogs: Micro-blog blogs are stored by rows, each blog includes tags, content and related features. Related features mainly include timestamp, number of comments, number of forwards and number of likes.

Word-bag: Word-bag is a set of keywords with an id. The keywords are from the text corresponding to the id and can describe the text. This paper mainly uses word-bag to describe text.

Heat: Heat is used to evaluate the popularity level, which is calculated by the related features of blog posts. Heat of blogs can be abbreviated as $Heat\left(d_i\right)$ and

Heat of event can be abbreviated as $Heat\,(E)$. Specific definitions are as follows,

$$Heat\,(d_i) = sum\,(features_i) = comments + forwards + likes \tag{1}$$

where $comments$ represents the number of comment, $forwards$ is the number of forwarding and $likes$ is the number of like in blogs. The sum of $Heat\,(d_i)$ represents the heat of the event $Heat\,(E)$.

3 Two-Stage Clustering Hot Event Detection Model

A two-stage clustering hot event detection model is proposed that contains both text-cluster and semantic-cluster. Because K-Means and LDA models will be used in each of two stages for improvement respectively, thus the proposed model is abbreviated as KMLDA.

3.1 Text-Cluster Stage

In text-cluster stage, micro-blog blogs are equally divided into slices to reduce the size of data. The blogs in each slice are divided into words by Jieba[1] and then converted to vectors using the Word2vec [10] method. Finally, K-Means is selected as a clustering method to cluster blogs of each slice. After text-cluster, many text clusters will be generated. Text clusters with a small number of blogs will be filtered because they represent insufficient discussion. Furthermore, K-Means is optimized to fit KMLDA model. Main optimizations are as follows.

(1) Cosine similarity is used as the measure distance of K-Means, and it is defined as Eq. (3),

$$\begin{aligned} dis &= 1 - \cos(\overrightarrow{w_i}, \overrightarrow{w_j}) \\ &= 1 - \frac{\overrightarrow{w_i} \cdot \overrightarrow{w_j}}{\|\overrightarrow{w_i}\| \cdot \|\overrightarrow{w_j}\|} \\ &= 1 - \frac{\sum\limits_{k=1}^{n} w_{i,k} \cdot w_{j,k}}{\sqrt{\sum\limits_{k=1}^{n} w_{i,k}^2}\sqrt{\sum\limits_{k=1}^{n} w_{j,k}^2}} \end{aligned} \tag{2}$$

where $\overrightarrow{w_i}$ represents weight vectors of blog d_i, n is feature dimension, and $w_{i,k}$ is the kth weight of blog d_i. When dis is smaller, the blog similarity will be higher.

(2) Set $AVG(dis)$ as a minimum similarity threshold. Because the accuracy of cluster-centers updating may be affected by large dis in the K-Means training process. Specific definition as Eq. (3),

$$AVG(dis) = \frac{SUM(dis_{max})}{NUM(d)} = \frac{1}{n} \cdot \sum_{i=1}^{n} dis_{max,i} \tag{3}$$

[1] "Jieba" (Chinese for "to stutter") Chinese text segmentation: built to be the best Python Chinese word segmentation module. GitHub: https://github.com/fxsjy/jieba/.

where $NUM(d)$ is the number of blogs, $SUM(dis_{max})$ is the sum of the cosine similarities $dis_{max,i}$ which are the maximum cosine similarity between text and cluster-centers.

$AVG(dis)$ as a threshold, dis beyond this threshold will not participate in update of cluster-centers.

3.2 Semantic-Cluster Stage

LDA model is chosen for semantic-cluster because it can accurately detect hot events when semantic information is sufficient. LDA model is implemented through the spark machine learning package. Meanwhile, how to input the results of text-cluster into LDA model is designed in detail. The process of processing text clusters is divided into two steps: keyword extraction and vector transformation.

Keywords are extracted from text clusters and used as word-bags. In order to find words of widespread concern, blog heat which is introduced in Sect. 2 is used to extract keywords.

With word-bags, we need to convert them to vectors because the input of LDA model is a vectorized text. TF-IDF method is used to vectorize word-bags due to high effective and adapted the characteristics of word-bags. Specific definitions are shown in Eq. (4),

$$w_{i,k} = tf_{i,k} * idf_k = \frac{n_{i,k}}{n_i} * \log(\frac{N}{n_k+1}) \tag{4}$$

where $tf_{i,k}$ represents word frequency, $n_{i,k}$ is the number of the kth word in word-bags wb_i, and n_i is the number of words in word-bags wb_i. idf_k represents reverse text frequency, N is the number of text clusters, n_k is the number of text clusters which include the kth word.$w_{i,k}$ represents weight of the kth word of word-bags wb_i.

After transforming word-bags into vectors by TF-IDF method, the vectors are used as input to cluster hot events by LDA model. The hot events clustered are displayed in the form of word-bags and sorted by $Heat\,(E)$ which is the heat of the event.

4 Parallel Computing Design and Implementation

Spark is a popular memory-based large data processing framework, which processes and stores data based on the data structure RDD. In order to meet the requirement of large-scale blog processing, a set of spark jobs are designed for KMLDA model so that to ensure event detection efficiently. Meanwhile, a parallel processing framework on Spark is designed for text-clustering to reduce the size of data per RDD and improve computational efficiency.

4.1 Updating Cluster-Center

The KMLDA framework is divided into two parts. The first part corresponds to the text-cluster stage. Firstly, micro-blog blogs and related features are read into RDD. Then, the RDD is equally divided into multiple parts which represent as $\{\mathrm{RDD}_1, \mathrm{RDD}_2,..., \mathrm{RDD}_\mathrm{n}\}$. For each $\mathrm{RDD}_i \in \mathrm{RDD}$, the clustering operation in Sect. 3.2 is executed by parallel. The second part corresponds to the semantic-cluster stage. All RDD which have processed by test-cluster are merged into one RDD. The operation in Sect. 3.3 and Sect. 3.4 are used to process this RDD and detect events.

4.2 Spark Implementation

In the text-cluster stage, it is necessary to update cluster-center when training K-Means. Training data is in one RDD makes it difficult to update cluster centers, because data within the RDD is difficult to interoperate. To solve this problem, a flag is added to each blog after judging cluster-center of the blog. The flag is used to mark which cluster center the blog belongs to. Flags will be grouped to update cluster-centers.

In the semantic-cluster stage, word-bags are needed to be transformed into vector by TF-IDF method. However, the calculation of IDF value is limited by the size of data. In order to efficiently calculate IDF value, an inverted sorting method is designed. Words are used as keys to cluster blogs and calculate their number. A Hashmap containing words and the number of texts is made. The Hashmap makes it easy to calculate IDF values.

5 Experiment and Analysis

5.1 Experimental Preparation

In order to verify accuracy and efficiency of KMLDA model, experiments are performed on Sina Weibo. Totally 49.19 million micro-blog blogs are collected by Sina Weibo API. The data have no specific category and longer than three words. Among them, 17 million micro-blog blogs are marked with a single word, like cooking, football, Messi, Trump, etc. The other data has type labels, like weather, sports, life, etc. This part of data is used to train KMLDA model and marked data is used to verify the accuracy of KMLDA model.

Test environment is a Spark cluster which has two nodes, each node is CentOS7 and 256 GB memory. Spark-LDA [2] and TMHTD [1] are chosen as comparative models. Spark-LDA improves LDA model to run in the Spark environment. TMHTD is an event detection model with two-layer cosine clusters which running in the Spark environment.

5.2 Accuracy Evaluation

Recall rate, accuracy rate and event accuracy rate are used as evaluation indicators. Specific definitions are as follows,

$$recall = \frac{N_{reality}}{N_{all}} \tag{5}$$

$$\text{accuracy} = \frac{N_{right}}{N_{reality}} \tag{6}$$

$$\text{eventAccuracy} = \frac{E_{right}}{E_{all}} \tag{7}$$

where $N_{reality}$ represents the number of blogs after clustering, N_{right} represents the number of blogs which are correctly clustered, E_{all} represents the number of blogs before clustering, E_{right} represents the number of events which are correctly detected, E_{all} represents the number of events.

These indicators accuracy rate and recall rate are based on blog, and eventAccuracy rate is based on event. The marked blogs are extracted to different sizes for accuracy verification, results are shown in Table 1.

Table 1. Accuracy evaluation.

Data size	Methods	Accuracy	Recall	EventAccuracy
64 MB	Spark-LDA	0.7832	0.8912	0.94
	TMHTD	0.8575	0.9131	0.94
	KMLDA	0.8523	**0.9254**	**0.96**
128 MB	Spark-LDA	0.7551	0.8543	0.90
	TMHTD	0.8564	0.8856	0.93
	KMLDA	0.8357	**0.9133**	**0.96**
512 MB	Spark-LDA	0.7324	0.8102	0.85
	TMHTD	0.8365	0.8772	0.89
	KMLDA	0.8336	**0.8935**	**0.93**
2 GB	Spark-LDA	0.6567	0.7154	0.75
	TMHTD	0.7886	0.8225	0.85
	KMLDA	0.7552	**0.8543**	**0.91**
4 GB	Spark-LDA	0.5546	0.6625	0.63
	TMHTD	0.7138	0.7856	0.81
	KMLDA	0.6958	**0.8127**	**0.86**

The experiment uses data sets under different size, including 64 MB, 128 MB, 512 MB, 2 GB and 4 GB. As can be seen from Table 1, the indicators show a downtrend with the increase of data size. Compared with TMHTD and KMLDA, the downtrend of Spark-LDA is obvious. In addition, TMHTD is slightly better than KMLDA in accuracy rate, because TMHTD is supposed to calculate cosine similarity between any two blogs. For the recall rate, KMLDA is higher than the other models. The reason is that KMLDA is not strict in setting blog filtering conditions. Meanwhile, KMLDA considers keywords extraction according to the heat of blogs, and LDA model has better event detection ability, so that KMLDA performs better than Spark-LDA and TMHTD in eventAccuracy rate.

5.3 Running Time Comparison

Time efficiency verification is mainly divided into running time comparison and scalability verification. As shown in Fig. 1.

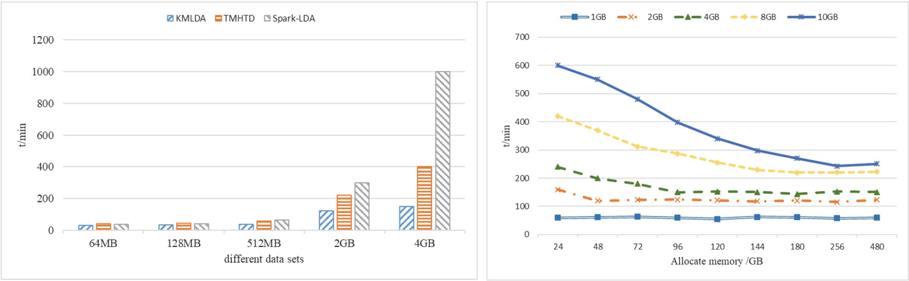

Fig. 1. Time efficiency evaluation.

The left figure represents the running time of each algorithm under different size of data. From figure, KMLDA has a significant improvement in running time which compared with Spark-LDA and TMHTD. It is proved that KMLDA which designs parallel calculation framework and linear algorithm complexity has high efficiency to detect events.

As shown in right figure, the scalability of KMLDA is tested by increasing memory and adjusting the size of data. From figure, the running time of KMLDA gradually decreases with the increase of memory size. However, by algorithm complexity, CPU resources and IO stream, the running time finally approaches to a stable value.

6 Summary

In this paper, a two-stage clustering hot event detection model KMLDA for micro-blog is proposed on Spark. This model considers the characteristics of blogs and time efficiency in big data environment. The process of KMLDA is divided into two stages. In text-cluster stage, data size can be reduced via slicing, and K-Means is adapted to improve the accuracy by threshold setting and cosine similarity. In semantic-cluster stage, word-bags are extracted from text clusters and then LDA model clusters word-bags to detect hot events. Experimental results show that KMLDA improves the accuracy and time efficiency of hot event detection in big data environment. In future work, how to integrate user characteristics and topic types to satisfy personalized event detection, and real-time data processing of micro-blog blogs will be considered.

Acknowledgments. This work was financially supported by the Natural Science Foundation of China (41571401).

References

1. Ai, W., Li, K., Li, K.: An effective hot topic detection method for microblog on spark. Appl. Soft Comput. **70**, 1010–1023 (2018)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
3. Cao, J.X., Xu, S., Chen, G.J., Zhao, L.Y., Zhou, T., Liu, B.: Discovering geographical topics in online social networks. Chin. J. Comput. **40**(7), 1530–1542 (2017)
4. Chen, X., Zhou, X., Sellis, T., Li, X.: Social event detection with retweeting behavior correlation. Expert Syst. Appl. **114**, 516–523 (2018)
5. Hao, Y., Zheng, Q., Chen, Y., Yan, C.: Recognition of abnormal behavior based on data of public opinion on the web. Comput. Res. Dev. **53**(3), 611–620 (2016)
6. Huang, F.L., Feng, S., Wang, D.L., Yu, G.: Mining topic sentiment in microblogging based on multi-feature fusion. Chin. J. Comput. **40**(4), 872–888 (2017)
7. Huang, F.L., Yu, G., Zhang, J.L., Li, C.X., Yuan, C.A., Lu, J.L.: Mining topic sentiment in micro-blogging based on micro-blogger social relation. J. Softw. **28**(3), 694–707 (2017)
8. Kitajima, R., Kobayashi, I.: A latent topic extracting method based on events in a document and its application. In: Proceedings of the ACL 2011 Student Session, pp. 30–35. Association for Computational Linguistics (2011)
9. Mathioudakis, M., Koudas, N.: TwitterMonitor: trend detection over the twitter stream. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp. 1155–1158. ACM (2010)
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Comput. Sci. (2013)
11. Ozdikis, O., Senkul, P., Oguztuzun, H.: Semantic expansion of hashtags for enhanced event detection in Twitter. In: Proceedings of VLDB 2012 Workshop on Online Social Systems, pp. 1–6 (08 2012)
12. Stilo, G., Velardi, P.: Temporal semantics: time-varying hashtag sense clustering. In: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (eds.) EKAW 2014. LNCS (LNAI), vol. 8876, pp. 563–578. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13704-9_42
13. Sun, R., Guo, S., Ji, D.H.: Topic representation integrated with event knowledge. Chin. J. Comput. **40**(4), 791–804 (2017)
14. Wang, Z.H., Chen, S.M., Yuan, X.R.: Visual analysis for microblog topic modeling. J. Softw. **29**(4), 1115–1130 (2018)
15. Xu, K., Qi, G., Huang, J., Wu, T., Fu, X.: Detecting bursts in sentiment-aware topics from social media. Knowl.-Based Syst. **141**, 44–54 (2018)
16. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1445–1456. ACM (2013)

17. Yilmaz, Y., Hero, A.O.: Multimodal event detection in Twitter hashtag networks. J. Signal Process. Syst. **90**(2), 185–200 (2018)
18. Zhong, Z.M., Guan, Y., Li, C.H., Liu, Z.T.: Localized top-k bursty event detection in microblog. Chin. J. Comput. **41**(7), 1504–1516 (2018)