2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
Banff Center, Banff, Canada, October 5-8, 2017

Topic detection Based on Similar Networks

Xin Liu
and Feng Wang
and Zhongwei Li
and Weishan Zhang
and Shuai Cao
China University of Petroleum (East China),
Qingdao, China 266500
Email: lx@upc.edu.cn

Sumi Helal
InfoLab21,
School of Computing and Communication
Lancaster University, UK
Email: s.helal@lancaster.ac.uk

Jiehan Zhou
University of Oulu,
Oulu, Finland
Email: jiehan.zhou@oulu.fi

***Abstract*—Social data from online social networks is expanding rapidly as the number of users and articles posted increases, making public opinion analysis a greater challenge. Real-time topic detection is a key part of public opinion analysis. The complex data processing involved in traditional clustering and text categorization can lead to time delays in topic detection. In this paper we construct similar networks and detect topics from similar communities that reduces the processing overhead in obtaining real-time topics. The similar communities consist of users with high similarity between them. We collect public topics from the microposts of delegates selected from each similar community. Selecting delegates can reduce the processing time of large amounts of redundant data during topic detection. Obtaining public opinion keywords in real time allows organizations to respond to public opinion security incidents in real time. Experiments showed that our scheme can find public topics faster and more effectively than two traditional algorithms.**

I. Introduction

The number of social network users is expected to reach some 2.95 billion worldwide by 2020 (around a third of Earth's population), with 650 million in China alone[1]. Online social networks are the main medium for public opinion because they offer real-time communication and convenience[2]. Microblogging is popular because the content is typically smaller in both actual and aggregated file size. Criminals or malicious users can distribute many false, adverse, or negative statements via microblogging networks, which threatens national security and social stability[3], [4]. Real-time response to hot public events to limit dissemination of such statements is vital, and is an ongoing area of focus for government, enterprises, and research institutions.

Statistics from social big data show that hundreds of millions of users post tweets every day on Twitter, the most popular social network [5]. It is difficult for researchers to rapidly analyze massive amounts of social network data [6], monitor and track public topics, and then support managers in responding to public safety incidents in a timely manner in a big data environment [7], [8], [9]. Currently, public opinion analysis mainly collects and stores social network data and combines it with text-mining techniques [10], such as clustering algorithms [11], to identify topics from a large volume of low-value density data [12]. With the proliferation of social big data, it is difficult for clustering algorithms, such as single-pass and k-means clustering algorithms, to obtain accurate central results in a timely manner. Single-pass algorithms rely on the input sequence of documents, so it is difficult to obtain accurate clustering results with miscellaneous disordered social big data. Low-value density social big data contains a great deal of noise and outliers, which can affect k-means clustering results. Moreover, existing algorithms cannot manipulate social big data quickly or detect topics in real time. Therefore, a major challenge in topic detection is how to concentrate massive data, extract valuable information from social big data, and then realize the internal correlations among associated data [13], [14], [15].

In this paper, we propose a topic-detection scheme based on community detection that can lower the time complexity. We firstly construct a similar network defined in Section 3.1 by analyzing the micropost data of users in a microblog network. Next, we detect the network community based on the similar network using the similarity degree between users. Finally, we select several users in each similar community to be delegates of similar users. We collected keywords from the microposts of delegates to identify public topics. Our scheme can reduce the time spent processing a large volume of redundant data, with no close correlation to the data expansion.

This paper is organized as follows. Section 2 provides an overview of related work. In Section 3, we describe similar networks and similar communities. The topic detection based on similar network is illustrated in Section 4. Our experiments and evaluation are presented in Section 5. Finally, we conclude our work and look into the future in Section 6.

II. Related works

Topic detection is the discovery of new topics in big social data [16]. Topic detection has been divided into two categories: retrospective event detection (RED) and new event detection (NED) [16], [17]. The literature classifies the main topic-detection algorithms into three categories: traditional clustering algorithms, algorithms based on the probabilistic model, and algorithms based on the graph model.

978-1-5386-1645-1/17/$31.00 ©2017 IEEE

A. Traditional algorithms for topic detection

Clustering algorithms [11], [18] have been used mainly for RED and NED tasks. RED focuses on discovering previously unidentified events from accumulated historical collections. The iterative clustering algorithms used in RED require the entire document collection to be organized in topic clusters. Hierarchical clustering approaches [19] have also been employed for this task. However, the computational complexity is high, and the clustering results can be impacted with singular value. The traditional k-means algorithm and its variants, which all use clustering algorithms, such as k-median and k-means++, have been applied widely [20], [21]. However, the clustering results can be impacted by the number of clusters k given beforehand, noise, and outliers. NED is the discovery of new events from live streams in real time. The clustering approaches that NED employs are typically based on incremental algorithms. The single-pass clustering algorithm [22] is a widely employed incremental algorithm. However, single-pass clustering results have the shortcoming of relying on the input sequence of documents.

Other traditional algorithms include the K-nearest-neighbor (KNN) algorithm [23], support vector machines (SVM) algorithm [24] and self organizing map (SOM) neural network clustering algorithm [25]. KNN is a concise and nonparametric algorithm based on learning by analogy. It is not suitable for big data because of its higher computational complexity, and the clustering result can be impacted by initial cluster center. SVM is a method for simultaneous multitopic discovery. But, the algorithm is relatively complex and its training speed is slow. The SOM neural network is developed by simulating the processing characteristics of the human brain. It is a type of artificial neural network used for clustering. The SOM clustering result can be impacted by the distortion produced when the input data is mapped to low-dimensional space.

B. Algorithms based on the probabilistic model

The famous topic model latent Dirichlet allocation (LDA) [26] is one of two probabilistic models proposed for topic detection [27]; the other is probabilistic latent semantic analysis (pLSA). Later, many variants based on the LDA were proposed. Wang et al. estimated probabilities of topic transition and predicted future topics from past observations using Temporal-LDA(TMLDA) [28]. Using LDA and matrix factorization, Kim et al. identified common and discriminative topics via joint nonnegative matrix factorization [29]. Yuan et al. presented light LDA, which enables very large data sizes and models to be processed on a small computer cluster [30]. Hu et al. captured both strength and content evolution simultaneously via online LDA to explore the evolution of development topics [31].

Because document-level word co-occurrences are sparse, the LDA model is not suitable for clustering short articles. Social big data includes many short articles. Selecting the number of topics and the sequence of words in the documents are also problems with the LDA model. Yin and Wang proposed a collapsed Gibbs sampling algorithm for the Dirichlet multinomial mixture model (GSDMM) for short text clustering that can cope with the sparse and high-dimensional problem of short texts [32], [33]. Maurus and Plant presented SkinnyDip, which is highly noise robust, practically parameter free, and completely deterministic [34]. It employs insightful recursion based on 'dips' into univariate projections that can detect a range of cluster shapes and densities [43]. The most significant problem is that the computation is huge when applied to social big data, , because each clustering document needs Dirichlet processing.

C. Algorithms based on the graph model

The graph analytical method is another algorithm used mainly for detecting topics. Ohsawa et al. constructed KeyGraph using terms and co-occurrence relations, and partitioned the graph for automatic indexing [35]. Sayyadi and Raschid applied the KeyGraph analytical approach to topic detection [36]. However, they did not use the semantic information derived from the topic model and measured co-occurrence relations in an isolated term, which reduced accuracy. Chen et al. considered the entire heterogeneous network to present a nonparametric heterogeneous graph scan (NPHGS) for event detection [37]. However, they focused on bursty event detection and identifying major events from data, which is not suitable for detecting real-time topics from social data. Zhang et al. leveraged a hybrid relations analysis approach to fuse semantic relations and co-occurrence relations into a term graph and detected topics from the graph [38]. The complexity of the algorithm is high because of fusing semantic relations and co-occurrence relations inspired by KeyGraph.

The proliferation of social big data increases the challenge for existing topic-detection algorithms, especially traditional algorithms, due to the time overhead and accuracy. To solve existing problems, we propose a topic-detection scheme based on similar network. Instead of clustering documents or keywords, we cluster publishers of microposts and discover topics from delegates. This change makes the similar users together in a community. Getting topics from a small number of delegates reduces the need to process large volumes of redundant information, making our scheme resilient to the data explosion.

III. Similar networks

To detect public topics from similar communities, we need to first construct a similar network. In our algorithm, a similar network is a reconstructed network of users and the similarity between users, rather than the original social network.

Similarity defines a relationship between users with the same keywords on their social network profiles and in their microposts. The number of same keywords indicates the degree of similarity, or *similarity distance*, between the users. The larger the similarity distance between two users, the more similar they are.

Users and the similarity between users constitute the *similar network*. We let nodes represent users. If the similarity distance between two users is greater than 0, we add a link between

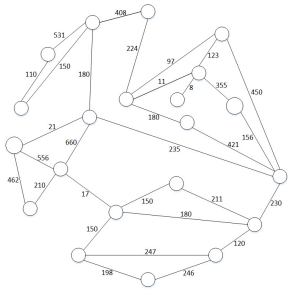

Fig. 1. A simple similar network.

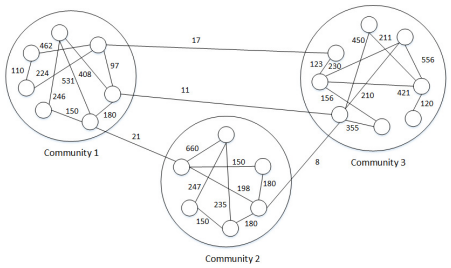

Fig. 2. Similar communities in the similar network.

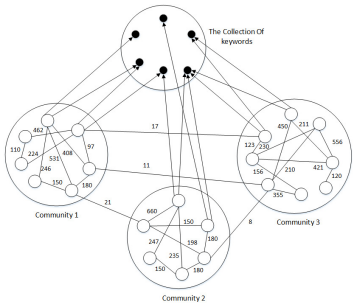

Fig. 3. The process of finding topics.

the two nodes. So we get an undirected graph of the similar network. The weight of each edge is marked by the similarity distance. Then, a graph of a simple similar network is formed, as shown in Fig. 1. We can detect the similar community based on this similar network.

IV. Topic-detection scheme based on similar network

Our method consists of four steps. The first step is defining the similarity between users formally. Secondly, the similar network graph can be constructed by users and the similarity between users. Next, we detect similar communities in the similar network. Finally, we collect the keywords from the microposts of delegates.

A. Defining the similarity

Users have different interests and focuses. We extract user interest keywords from the content of their microposts during six days. These keywords are sorted to form TOP_k keywords, where k is the number of all keywords obtained and will be updated with the keywords updating. To calculate the similarity between nodes, we set a keyword vector. The keyword vector for each node is derived from its interest keywords from its microposts according to the TOP_k keywords. The keyword vector of user i is represented by $Vi = [b_1 b_2 \dots b_k]$, where b_i is 1 or 0, i = 1, 2, …, k, with 1 indicating that the corresponding keyword is one of the user's interests in the TOP_k keywords, and 0 indicating it is not the user's interest.

B. Constructing the similar network

The similarity distance between two nodes is the number of the same keywords between these two users. According to the keyword vectors of any two users, their similarity distance can be calculated using (1).

$$H_{ij} = count(V_i AND V_j) \tag{1}$$

where H_{ij} is the similarity distance between node i and node j. The function count is used to get the number of 1 in the result vector which indicates the same interests between node i and node j.

C. Detecting the similar community

We detected similar communities using a clustering algorithm. Initially, each node is a separate community. We set the value of the similarity threshold in advance. Then we randomly select one node in a community. If the similarity distance between a node and the select node is over the threshold, the node should be added to the community where the selected one is. The process should be carried out for all other nodes except the selected one. After that, we should select another separate node to do that until there isn't separate node. The similar nodes cluster into a mutual community called a similar community. The result is shown in Fig. 2.

The similar community detected includes those nodes that have similar interests and focuses. In the similar community, users might care about the same events and have similar opinions.

D. Finding topics

We select n nodes as the delegates of nodes in each similar community and collect the keywords, as shown in Fig. 3.

In each community, first, we select a node randomly. Then, we select other $n-1$ nodes that have lower similarity with the selected nodes. We obtain the real-time topics from the keywords in new microposts of these n nodes. We make statistics and rank all real-time keywords. The top keywords can indicate the real-time topics.

The smaller the similarity distance between two nodes, the greater the difference between the keywords attached to the two users. This is why we obtain n different nodes that have lower similarity distances with each other. We can get more different keywords in the mutual community to avoid missing

important topics. To find public topics accurately, the value of n should be pre-determined before getting the real topics.

After detecting communities, we get m similar communities denoted as $C_1, C_2, ..., C_m$. C_i is the ith community, where $i = 1, 2, ..., m$. We can get a keyword set which includes all the keywords in each similar community. M_i and P_i are the keyword set and the node set of C_i separately, where $i = 1, 2, ..., m$. First, we randomly choose one node as a delegate from C_i which denoted as $Node_1$, and add $Node_1$ into P_i . We obtain all the keywords from $Node_1$ as keyword set K_1, $K = K \bigcup K_1$, if $K = M_i$, then stop selecting delegate node; if not, we select another nodes $Node_2$ that have lower similarity with $Node_1$, repeat the above steps until $K = M_i$. then we get the number of nodes in P_i. For all similar communities, we let

$$n = MAX(|P_1|, |P_2|, ..., |P_m|) \tag{2}$$

For each similar communities, we select n delegates, so the keyword set of all the delegates in a similar community is the same as the keywords of the similar community. All the keywords in the similar network can be obtained from the delegates in all similar communities.

V. Experiments and Evaluation

User data from the Sina microblog is a good candidate for public opinion analysis. Using a crawler, we collected raw microblog data from 8/14/2016 to 8/20/2016 for our experiments. TOPk keywords of topics in our experiments were from the collected microblog data. During this period, we also got the top 13 public topics in China from Baidu and Sina [39], [40]. We assessed the accuracy of topic detection according to the actual top public topics in our experiments.

A. Experimental hypothesis

Our experiments are based on the following assumptions.

i. The accuracy of topic detection is the ratio of correct topics found through our experiments to all actual topics.

ii. The average similarity distance among users may be different due to the closeness among nodes in each social network, thus we should set the threshold of community detection according to the specific network environment. In our experiments, we calculate the threshold of community detection Y using (2).

$$Y = S_{min} + \alpha \times (S_{max} - S_{min}) \tag{3}$$

where S_{min} is the minimum similarity between nodes in the similar network; S_{max} is the maximum; α is the coefficient of community detection. Its value can be determined in our experiments, $\alpha \in [0, 1]$.

iii. The noise is words that are in the user's article but aren't relevant to its content.

iv. If the number of nodes in a similar community is less than the predetermined number of delegates, we will not select delegates from the community.

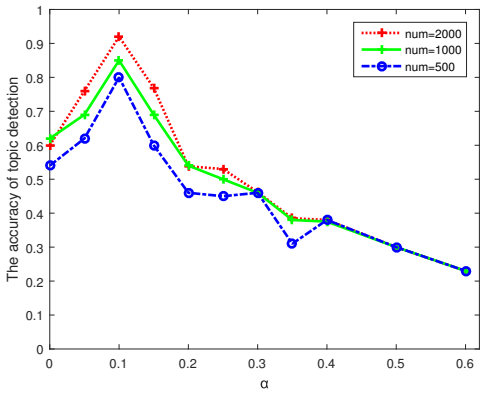

Fig. 4. Accuracy for different community-detection coefficients.

B. Experiments

We obtained an average of 20 blog articles in a random way reflecting each user's opinion, giving us 40,000 web documents from 2,000 users which can be representative of this social network to use in our experiments because of the information redundancy. Our experiments are divided into the following cases according to several influential factors.

1) The coefficient: We chose five delegates from every community and conducted experiments with 500, 1,000, and 2,000 nodes in the social network separately. The results are shown in Fig. 4. The community-detection coefficient affects topic detection dramatically. There are similar peaks at the same point (0.1) in the three curves in Fig.4, indicating that it is suitable for 0.1 to be chosen as the coefficient regardless of the scale of the social network. We can take 0.1 as the value of the coefficient for community detection in a social network such as the network in our experiments.

2) Comparison with other algorithms: We compare our scheme with both a single-pass algorithm, a k-means algorithm and LDA. For these three classical algorithms, we treat each blog article as a document. We compared the three algorithms with our scheme using the same microblog data. We identified the top seven topics according to the data from the three methods in our experiments, as shown in Table 1.

Obviously, our scheme found more topics, which indicates that our scheme is superior to the other two algorithms in regard to topic-detection accuracy.

3) Scale of the similar network: To test the similar network's scalability, we constructed simulated similar networks with 2,000, 3,000, 5,000, 8,000, and 10,000 nodes, respectively. We set the appropriate n with every network in advance. The accuracy of topic detection under different scales of the similar network is shown in Table 2. As the table shows, the accuracy of our scheme is basically unchanged as the number of nodes increases, indicating that our scheme isn't affected by the scale of the social network.

4) Noise: In our experiments, we artificially increase the number of irrelevant keywords to evaluate the performance of our scheme. We conduct experiments in the network with

TABLE I
TOPIC-DETECTION RESULTS FOR THREE ALGORITHMS.

Topic	Single-pass	K-means	LDA	Our Scheme
Rio Olympics	√	√	√	√
Wang Divorce	√	√	√	√
The Girl Fu	√	√	×	√
Teenager Burning Teacher	√	√	√	√
Female Teacher Expelled Due to Cancer	×	√	√	√
Subway Staff Swear	√	√	√	√
Tampering the Application Form of College	√	√	×	√
Southern Flood	√	√	√	√
Fight Against Flood	×	×	×	√
Arbitration in South China Sea	√	√	√	√
Peking University Dean Interview Learning Tyrants	×	×	√	√
Teacher Suspended Boy to Death Because Eating Snacks	√	×	√	√

TABLE II
THE ACCURACY OF TOPIC DETECTION WITH THE DIFFERENT NUMBER OF NODES.

The number of nodes	2000	3000	5000	8000	10000
Accuracy	92.31%	92.85%	93.33%	92.31%	93.33%

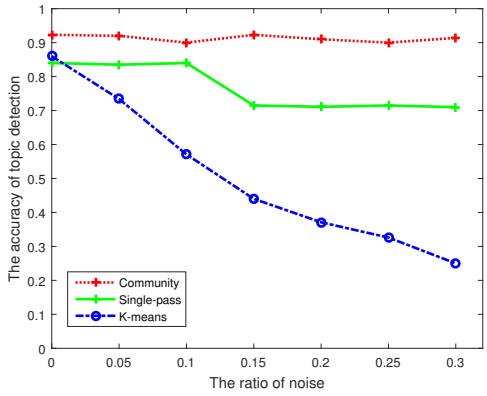

Fig. 5. The effects with the different ratio of noise.

single-pass, k-means, and our scheme separately. According to experiments (1) and (2), we set the coefficient α to 0.1, the number of delegates n to 5 with our scheme. Figure 5 shows the results. The noise has no effect on our scheme basically because the accuracy doesn't decrease as the ratio of noise increases. The effect of noise on the single-pass algorithm is not large. Because the noise affected one cluster of topics, the accuracy decreased when the ratio of noise is 15 percent. Noise has the greatest effect on the k-means algorithm: when the ratio of noises increases, the accuracy declines sharply.

5) Detecting real-time topics: We detected the topics from microposts of these delegates at 10:00 a.m. on 9/6/2016 GMT+8. We integrated Baidu and Sina's real-time hot list, and obtained the top 10 hotspots at 10:00 a.m. The top 10 topics were the same as our topics: *Xi Back to Alma Mater, Apple's Official Website Paralysis, Qvod Company Pleaded Guilty in Court, Mainland Tourists Reducing, Villa Waste, Restaurant Explosion, Forced Daughter Married, Set Six Thousand Sheep Free, Harassment of Female Tenant and Expired Drugs for Surgery*. This experiment indicated that the construction of the similar communities was almost unchanged over 20 days.

The public topics can be updated by collecting keywords from these delegates whenever necessary. Without the clustering and community-detection process, this process needs very little time overhead. The community detection and delegates chosen can be updated periodically which can adapt to network and user changes. The results show our scheme is effective and can provide the hottest topics in real time.

C. Time complexity analysis

We compared the time cost of our scheme with other two algorithms by analyzing their time complexity.

The time complexity of the single-pass and k-means algorithms is $O(n^2)$, where n is the number of documents.

For the social network with m nodes, we obtained c microblog articles from each node to form the set of documents, where c is a constant. Therefore, $n = c \times m$. In the first step, we extract keywords from each node's microblog articles and calculate the similarity distance. The corresponding time complexity is $O(m)$ and $O(m^2)$ separately. In the community-detection step, the time complexity of community detection based on clustering is $O(m)$. In the step of finding topics, we get the keywords from the delegate of each community. Thus, the time complexity is related to the number of communities. It is almost constant for collecting keywords at a certain moment. That is, the time complexity of finding topics is $O(1)$.

Community detection is compiled in the background in advance and updated periodically. To detect real-time topics, we only obtain keywords from delegates of each similar community detected previously. That is, the time complexity is $O(1)$.

VI. CONCLUSION

In this paper, we proposed a topic-detection scheme based on similar network. We defined the similarity between users by analyzing the microblog article data of social network users. Based on the similarity, we detected the similar communities

in the similar network. Finally, we chose delegates of users from similar communities, and obtained public topics from the delegates. We concentrate the massive social data to avoid having to process a great deal of redundant data through selecting a small number of delegates from similar communities. The time complexity of getting real-time topics is $O(1)$.

Topic tracking and prediction are other challenges in the advent of social big data [41], [42]. In the future, we will track the development process of hotspot events and predict public topics within the network community structure.

ACKNOWLEDGMENT

The work presented in this paper is supported by National Natural Science Foundation of China (No. 61309024), the Key Program of Shandong Province (No. 2017GGX10140), Shandong Provincial Natural Science Foundation (No. ZR2015FM022), and the Scientific Research Foundation of China University of Petroleum (No. Y1207013).

REFERENCES

[1] Statista, *Statistics and facts about Social Networks*, https://www.statista.com/topics/1164/social-networks/.

[2] F. W and G. M. D, "The power of social media analytics," *Communications of the ACM*, vol. 57, no. 6, pp. 74–81, 2014.

[3] W. X, G. M. S, and B. D. E., "Automatic crime prediction using events extracted from twitter posts," in *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*. Springer Berlin Heidelberg, 2012, pp. 231–238.

[4] T. A, S. T. O, S. P. G, and et al., "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Fourth International AAAI Conference on Web and Social Media*. Association for the Advancement of Artificial Intelligence, 2010, pp. 178–185.

[5] I. L. Stats, *Twitter Usage Statistics*, http://www.internetlivestats.com/twitter-statistics/.

[6] H. J and W. M. L, "Searching twitter: Separating the tweet from the chaff," pp. 161–168, 2011.

[7] B.-M. F, M. M, and P. B, "Meta-level sentiment models for big social data analysis," *Knowledge-Based Systems*, vol. 69, pp. 86–99, 2014.

[8] C. Z and L. B, "Mining topics in documents: Standing on the shoulders of big data," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 1116–1125.

[9] C. M, M. S, and L. Y, "Big data: a survey," *Mobile Networks and Applications*, vol. 19, pp. 171–209, 2014.

[10] H. H, H. A, and M. H, "Selection criteria for text mining approaches," *Computers in Human Behavior*, vol. 51, pp. 729–733, 2015.

[11] Y. Q and D. D, "Text detection and recognition in imagery: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1480–1500, 2015.

[12] K. H. P, K. P, and Z. A, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, pp. 337–348, 2009.

[13] A.-Y. S, A. S, G. A, and et al., "Maqsa: A system for social analytics on news," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012, pp. 653–656.

[14] M. K and M. K. W, "Big data: New opportunities and new challenges [guest editors' introduction]," *Computer*, vol. 46, pp. 22–24, 2013.

[15] B. H, I. D, N. M, and et al., "Identifying content for planned events across social media sites," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*. ACM, 2012, pp. 533–542.

[16] A. J, C. J. G, D. G, and et al., "Topic detection and tracking pilot study final report," in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 194–218.

[17] Y. Y, P. T, and C. J, "A study of retrospective and on-line event detection," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1998, pp. 28–36.

[18] G. S, M. A, M. N, and et al., "Clustering data streams: Theory and practice," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, pp. 515–528, 2003.

[19] N. H. L, W. Y. K, and N. W. K, "A survey on data stream clustering and classification," *Knowledge and Information Systems*, vol. 45, pp. 535–569, 2015.

[20] L. A, V. N, and V. J. J, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, pp. 451–461, 2003.

[21] J. A. K, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, pp. 651–666, 2010.

[22] G. S, M. N, M. R, and et al., "Clustering data streams," in *proceedings 41st Annual Symposium on Foundations of Computer Science, 2000.* IEEE, 2000, pp. 359–366.

[23] J. S, P. G, W. M, and et al., "An improved k-nearest-neighbor algorithm for text categorization," *Expert Systems with Applications*, vol. 39, pp. 1503–1509, 2012.

[24] F. T and J. T, "Supervised clustering with support vector machines," in *ICML '05 Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005, pp. 217–224.

[25] I. D, K. V. P, and L. L. H, "Using the self organizing map for clustering of text documents," *Expert Systems with Applications*, vol. 36, pp. 9584–9591, 2009.

[26] B. D. M, N. A. Y, and J. M. I, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[27] D. Q, G. F, and Z. Y, "Early detection method for emerging topics based on dynamic bayesian networks in micro-blogging networks," *Expert Systems with Applications*, vol. 57, pp. 285–295, 2016.

[28] W. Y, A. E, and B. M, "Tm-lda: Efficient online modeling of latent topic transitions in social media," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 123–131.

[29] K. H, C. J, K. J, and et al., "Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 567–576.

[30] Y. J, G. F, H. Q, and et al., "Lightlda: Big topic models on modest computer clusters," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 1351–1361.

[31] H. J, S. X, and L. B, "Explore the evolution of development topics via on-line lda," in *Proceedings 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. IEEE, 2015, pp. 555–559.

[32] Y. J and W. J, "A dirichlet multinomial mixture model-based approach for short text clustering," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 233–242.

[33] ——, "A text clustering algorithm using an online clustering scheme for initialization," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1995–2004.

[34] M. S and P. C, "Skinny-dip: Clustering in a sea of noise," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1055–1064.

[35] O. Y, B. N. E, and Y. M, "Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor," in *Proceedings IEEE International Forum on Research and Technology Advances in Digital Libraries, ADL 98*. IEEE, 1998, pp. 12–18.

[36] S. H and R. L, "A graph analytical approach for topic detection," *ACM Transactions on Internet Technology*, vol. 13, pp. 992–999, 2013.

[37] C. F and N. D. B, "Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1166–1175.

[38] Z. C, W. H, C. L, and et al., "A hybrid termterm relations analysis approach for topic detection," *Knowledge-Based Systems*, vol. 93, pp. 109–120, 2016.

[39] Baidu, *top.baidu*, http://www.top.baidu.com/.

[40] Sina, *top.weibo*, http://s.weibo.com/top/summary/.

[41] K. K and G. V, "A survey of topic tracking techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, pp. 383–392, 2012.

[42] A. F and K. W, "A survey of techniques for event detection in twitter," *Computational Intelligence*, vol. 31, pp. 132–164, 2015.