IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 3, 2020, accepted February 9, 2020, date of publication February 12, 2020, date of current version February 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973430

BTM and GloVe Similarity Linear Fusion-Based Short Text Clustering Algorithm for Microblog Hot Topic Discovery

DI WU, MENGTIAN ZHANG, CHAO SHEN, ZHUYUN HUANG, AND MINGXING GU
Department of Information and Electronic Engineering, Hebei University of Engineering, Handan 056038, China
Hebei Key Laboratory of Security Protection Information Sensing and Processing, Hebei University of Engineering, Handan 056038, China

Corresponding author: Chao Shen (hbgcdx123@163.com)

This work was supported in part by the Reaearch Projects of Science and Technology in Hebei Higher Education Institutions under Grant ZD2018087, Grant ZD2016017, Grant QN2018109, and Grant YQ2014014, in part by the Nature Science Foundation of Hebei Province under Grant F2019402428, in part by the National Key RD Program of China under Grant 2018YFF0301004, and in part by the National Natural Foundation of China under Grant 61802107.

ABSTRACT Microblog hot topic discovery is one of the research hotspots in the field of text mining. The distance function of traditional K-means leads to low clustering accuracy, which leads to poor hot topic discovery. Three definitions are proposed in this paper: title words and body words, positional contribution-based weight and fusion similarity-based distance. The short text clustering algorithm based on BTM and GloVe similarity linear fusion (BG & SLF-Kmeans) is further proposed. BTM and GloVe are used to model the preprocessed microblog short texts. JS divergence is adopted to calculate the text similarity based on BTM topic modeling. WMD of improved word weight (IWMD) is used to calculate the text similarity based on GloVe word vector modeling. Finally, the two similarities are linearly fused and used as the distance function to realize K-means clustering. Specific word sets of 6 hot topics can be obtained, and microblog hot topics can be discovered. The experimental results show that BG & SLF-Kmeans significantly improves clustering accuracy compared with TF-IDF & K-means, BTM & K-means, and BTF & SLF-Kmeans.

INDEX TERMS BTM, GloVe, microblog hot topic discovery, similarity linear fusion, WMD.

I. INTRODUCTION

Because the advantages of extensive content and fast dissemination, microblogs have become an important platform for understanding the market economy, current affairs and other information [1]. The Sina microblog is one of the largest social platforms in China, and millions of messages are posted to this platform every day. The Sina microblog has some official news media and Sina ‘‘big V’’. Here, Sina ‘‘big V’’refers to people or enterprises with certain popularity or influence. Usually, they post microblogs with topic tags (double ‘‘#’’ or square brackets). These microblogs are more influential and more likely to contain hot topics. Hot topics here refer to topics that have been discussed by the public more often in a certain period of time. Unlike twitter, Sina microblogs are mostly short in Chinese. By mining deep semantic information from these texts, users can obtain the latest hot topics, and network supervision departments can correctly guide public opinion [2]. How to accurately discover hot topics from a large number of microblogs is a significant research subject.

Different from traditional texts, microblogs are mostly short in length and lack contextual information [3]. Therefore, how to express microblog short texts to carry more semantic information is an urgent problem to be solved in the research of microblog hot topic discovery.

In the traditional methods, TF-IDF was used to extract features from microblog short texts, and a clustering algorithm was applied to further obtain hot topics. To solve the problem of high dimensionality and sparsity, Zheng *et al.* [4] proposed a short text oriented clustering method. Feature vectors of texts were generated by TF-IDF, and then K-means was applied to extract hot topics (this algorithm is named TF-IDF & K-means below.) Aiming at the problem that IDF cannot change dynamically with the dataset, Yan *et al.* [5] proposed an improved single-pass algorithm. The improved

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

TF-IDF was used to extract features from the preprocessed microblog short text set, and then the improved single-pass was used to detect microblog topics.

TF-IDF counts words based on term frequency and inverse document frequency, resulting in neglecting the semantic information of words [6]. Therefore, to make the vectorized texts contain semantic information, some scholars applied topic models to mine topics. Topic models are statistical models based on clustering. By clustering the implicit semantic structure of the text set, several groups of specific word sets can be obtained, and then several topics can be extracted from them. The topic here refers to a central phrase that can summarize the specific word set. For hot topic discovery, topic refers to a phrase or sentence that can summarize an event. In view of the short length and complex structure of microblog texts, Sun *et al.* [7] proposed a microblog hot topic detection algorithm based on two-stage clustering. The PLSA model and K-means were combined for secondary clustering to detect microblog hot topics. Zhou *et al.* [8] developed a tag generation model for the subtopics of public opinion events (ET-TAG). PLSA with Background Language Model (PLSA-BLM) [9] and a subtopic keywords clustering algorithm were successfully combined to find subtopics of events. Chen *et al.* [10] improved the LDA model and proposed a microblog hot topic detection model (FSC-LDA). Feature selection and text clustering were united to identify the number of topics adaptively and identify hot topics more accurately. Aiming at the problem that global and local topics are forced alignment in the cross-collection LDA (ccLDA) model [11], Chen *et al.* [12] proposed an improved ccLDA (ICCLDA) topic model. Each word was determined to belong to a global or local topic during sampling to reduce the dispersion degree of words and detect multisource topics. To solve the problem of differences in features between Chinese and English texts, Chen *et al.* [13] proposed an improved Chinese English LDA (ICE-LDA) model.

The traditional topic model is suitable for long texts but not for short texts. In 2013, Yan *et al.* [14] proposed a biterm topic model (BTM). BTM models the word cooccurrence patterns (i.e., biterms) explicitly, rather than implicitly (via document modeling), to enhance topic learning. Additionally, BTM uses the aggregated word cooccurrence patterns in the corpus for topic discovering, which avoids the problem of sparse patterns at the document level.

Then, many scholars applied BTM to discover topics. To overcome data sparsity and expression diversity, Feng and Fang [15] proposed a microblog hot topic discovery method based on BTM. Modeled by BTM, high-frequency words are extracted to obtain a high-frequency word matrix of the underlying topic. Then, VSM is applied to reduce the dimensions and highlight the main features of texts, thus achieving microblog hot topics. When the difference between topic documents is distinct, K-means can discriminate topics, so Li *et al.* [16] developed a microblog topic detection method based on BTM and K-means. The sparsity of microblog short texts was alleviated by BTM, and K-means was adopted to detect topics (this algorithm is named BTM & K-means below.) The user interaction-based bursty topic model (UIBTM) [17] was mined by Li *et al.* By considering the number of comments and likes, semantic information was enriched, and sparsity is effectively overcome. Yan *et al.* [18] proposed a bursty biterm topic model (BBTM). Burstiness of biterms was used as prior knowledge for bursty topic modeling, and high quality bursty topics were discovered automatically.

To improve the accuracy of topic clustering, some scholars combined text semantics and word frequency. For the problem that hidden information of texts may be ignored by the traditional method, Wang *et al.* [19] developed a text clustering algorithm based on LDA. The text similarity based on LDA and TF-IDF was calculated. The two similarities were linearly fused, and K-means clustering was performed. For the problem that traditional methods have low efficiency and accuracy for short text modeling, Wang and Hu [20] proposed a hotspot detection method in microblog public opinion based on BTM. Modeled by BTM and improved TF-IDF, the corresponding text similarity was calculated. Then, the two similarities were linearly fused, and the K-means algorithm was used to obtain hot topics (this algorithm is named BTF & SLF-Kmeans below.)

Since TF-IDF is based on statistics, and microblog texts are mostly short in length, the document-word vector matrix constructed by TF-IDF is sparse [21]. In 2013, word2vec was developed by Google to train word vectors, which has a certain dimensionality reduction effect. Word2vec has two modeling methods: CBOW and skip gram [22]. Guo *et al.* [23] proposed a LDA model-based topic detection method. CBOW was introduced to vectorize texts to effectively reduce dimensions, and LDA was then performed to discover topics. Lu *et al.* [24] proposed a verb-biterm topic model (V-BTM). Word2vec was adopted to vectorize verbs and distinguish intentions with verb clustering. Then, BTM was applied to mine topics on dataset without verbs.

Shortly after the word2vec model was proposed, global vectors for word representation (GloVe) [25] was developed by Pennington *et al.* Global matrix factorization was combined with a local context window, making the trained word vector carry more semantic information. Additionally, GloVe only trained on the nonzero elements in the word-word cooccurrence matrix, rather than on the whole sparse matrix, effectively alleviating the sparsity problem [26].

For the word vector, using traditional distance formulas, such as Jensen-Shannon divergence (JS divergence) and Kullback-Leibler divergence (KL divergence), only the similarity between the two word vectors can be calculated, and the similarity between the texts cannot be obtained. Word Mover's Distance (WMD) [27] was proposed by Kusner *et al.* It considers the sum of the shortest distance of specific word vectors in one short text flow to specific word vectors in another as the similarity between the two texts [28].

For the problem that the distance function of K-means affects clustering accuracy, the advantages of BTM in distinguishing meaning and GloVe in dealing with polysemy is combined, the short text clustering algorithm based on BTM and GloVe similarity linear fusion (BG & SLF-Kmeans) is proposed and applied to discover microblog hot topics. JS divergence is used to calculate the text similarity after BTM modeling. To improve the accuracy of similarity calculation, the WMD of the improved word weight (IWMD) is used to calculate the text similarity after GloVe modeling. To improve the distance function of K-means, the linear fusion of the two similarities is used to improve the clustering accuracy, and thus obtaining a better hot topic discovery effect.

The rest of the paper is organized as follows. Section II presents problem definitions. The implementation process of BG & SLF-Kmeans is introduced in detail in Section III. Experimental results and analysis are presented in Section IV. Section V discusses the conclusions.

II. PROBLEM DEFINITIONS

A. TITLE WORDS AND BODY WORDS

For a news microblog short text, the title is usually at the front and marked with double "#" or square brackets; the rest is the body part. The news title plays the role of summarizing the news content.

Definition 1 (Title Words and Body Words): Assuming that the first 10 words of any preprocessed news microblog short text are title words, the rest are body words. That is, if the column label of the word meets $l_s < 10$, then the word is a title word; otherwise, it is a body word.

B. POSITIONAL CONTRIBUTION-BASED WEIGHT

WMD is only measured by TF when calculating the weight transform cost of words. This method is relatively rough because there are some words with high frequency but little contribution to topic discovery. Therefore, it is difficult to accurately reflect the differences in words. Additionally, the importance of title words and body words is different. Therefore, the positional factor of words should also be considered, therefore, the positional contribution-based weight is proposed.

Definition 2 (Positional Contribution-Based Weight): The TF-IDF value of words is used to calculate the weight transform cost of words. The positional contribution of title words $\gamma_1 = 1.5$ and of body words $\gamma_2 = 1$. Some words may be both title words and body words. The positional contribution-based weight of words can be described as

$$w\_pc_s = \left(\frac{c_{s\_title}}{c_s} \times \gamma_1 + \frac{c_{s\_body}}{c_s} \times \gamma_2\right) \times tf_s \times idf_s \qquad (1)$$

where c_s represents the total number of occurrences of word, c_{s_title} represents the frequency that s is a title word, c_{s_body} represents the frequency that s is a body word, and $c_{s\_title} + c_{s\_body} = c_s$, tf_s and idf_s are calculated as

$$tf_s = \frac{c_s}{\sum_{t=1}^{G} c_t} \qquad (2)$$

$$idf_s = \log \frac{|D|}{1 + |\{\mathrm{i} : \mathrm{s} \in \mathrm{d}_i\}|} \qquad (3)$$

where G is the size of vocabulary, |D| represents the number of short text set, $|\{\mathrm{i} : \mathrm{s} \in \mathrm{d}_i\}|$ represents the number of texts containing word s.

Example: Assuming that there are 100 microblog short texts with 1,000 words, the word "fire"appeared 20 times in 9 short texts, 15 times in the title and 5 times in the body.

If TF is used to calculate the weight transform cost of words, then $w_{fire} = \frac{20}{1000} = 0.02$.

If positional contribution-based weight is used to calculate the weight transform cost of words, then $w\_pc_{fire} = (\frac{15}{20} \times 1.5 + \frac{5}{20} \times 1) \times 0.02 \times \log \frac{100}{1+9} = 0.0275$.

C. FUSION SIMILARITY-BASED DISTANCE

For clustering algorithms, it is very important to calculate the similarity between the text and each clustering center to judge the cluster to which the text belongs. Therefore, the selection of distance function plays an important role in the clustering effect, so the fusion similarity-based distance is proposed.

Definition 3 (Fusion Similarity-Based Distance): Assuming that the text similarity based on BTM topic modeling and JS divergence is $Dis_B(d_i, d_j)$, the text similarity based on GloVe word vector modeling and IWMD is $Dis_G(d_i, d_j)$, then the fusion similarity-based distance can be described as

$$\begin{aligned} Dis\_fs(d_i, c_e) &= \lambda \cdot Dis_B(d_i, c_e) + (1 - \lambda) \cdot Dis_G(d_i, c_e) \\ & i = 1, 2, \cdots, n; \quad e = 1, 2, \cdots, K \end{aligned} \qquad (4)$$

where d_i is a text in dataset $D = \{d_1, d_2, \cdots, d_n\}$, c_e is the cluster center, K represents the number of clusters, λ represents the fusion coefficient and $0<\lambda<1$. The value of λ is determined by the clustering effect.

III. BTM AND GLOVE SIMILARITY LINEAR FUSION-BASED SHORT TEXT CLUSTERING ALGORITHM FOR MICROBLOG HOT TOPIC DISCOVERY

Considering that the K-means distance function affects the clustering accuracy, the BG & SLF-Kmeans short text clustering algorithm is proposed and applied to discover microblog hot topics. Microblog short texts are collected and preprocessed. Then, they are modeled with BTM and GloVe. After BTM topic modeling, JS divergence is adopted to calculate the text similarity according to text representation. IWMD is used to calculate the text similarity after GloVe word vector modeling. Finally, the fusion similarity-based distance is applied to K-means to improve the clustering accuracy to improve the quality of hot topic discovery. The flowchart of BG & SLF-Kmeans algorithm is shown in Fig. 1.

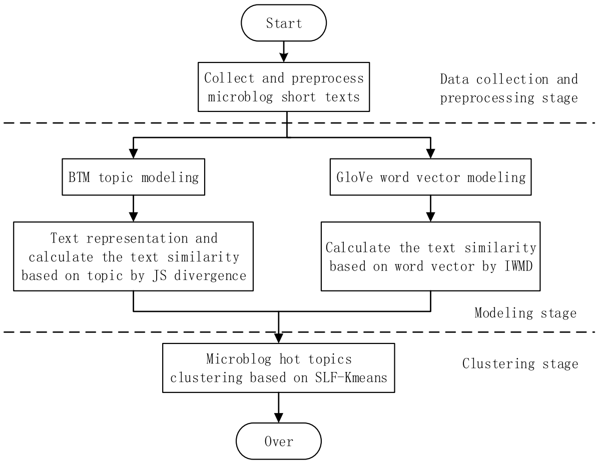
Start
Collect and preprocess microblog short texts
Data collection and preprocessing stage
BTM topic modeling
GloVe word vector modeling
Text representation and calculate the text similarity based on topic by JS divergence
Calculate the text similarity based on word vector by IWMD
Modeling stage
Microblog hot topics clustering based on SLF-Kmeans
Clustering stage
Over

FIGURE 1. The flowchart of BG & SLF-Kmeans algorithm.

A. MICROBLOG SHORT TEXT PREPROCESSING

Microblog short text preprocessing mainly includes four parts: microblog short text filtering, word segmentation and POS tagging, stop words removing, and feature selection. The specific process is shown in Fig. 2.

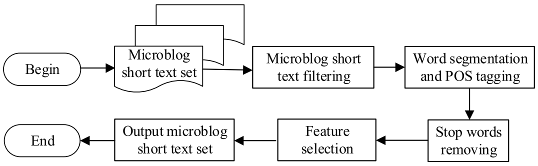
Begin
Microblog short text set
Microblog short text filtering
Word segmentation and POS tagging
Stop words removing
Feature selection
Output microblog short text set
End

FIGURE 2. The flowchart of microblog short text preprocessing.

After acquiring the microblog short text set, short text filtering is used to delete useless information such as emoticons, links, marker symbols, and ultra-short microblog texts with less than 10 words. Then, word segmentation and POS tagging as well as stop words removal are carried out. Finally, feature selection is performed. Some words contain less topicality, such as adjectives and adverbs. To improve the efficiency of the algorithm, only nouns and verbs are retained in the experimental data.

B. TEXT SIMILARITY MEASUREMENT BASED ON BTM TOPIC MODELING

The process of text similarity measurement based on BTM topic modeling is divided into two parts. The first part determines the optimal number of topics by calculating perplexity, then, BTM is used to model the preprocessed microblog short text set, and texts are represented according to the modeling results. The second part calculates the text similarity by JS divergence.

1) BTM TOPIC MODELING

Since the selection of topic number K directly affects the modeling results of BTM, it is necessary to determine K before modeling, which can make the modeling results optimal. The optimal value of K can be determined by calculating perplexity [29]. Perplexity is used to evaluate the generalization ability of the model; the smaller the perplexity, the better the modeling effect. The formula of perplexity is as follows:

$$\text{perplexity} = \exp\{-\frac{\sum \ln p(b)}{|\mathrm{B}|}\} \tag{5}$$

where |B| is the total number of biterm, $p(b)$ is the joint probability of biterm, and its formula is as follows:

$$p(b) = \sum_{z} p(z)p(w_i|z)p(w_j|z) = \sum_{z} \theta_z \phi_{i|z}\phi_{j|z} \tag{6}$$

where $P(z) = \theta_z$ represents the probability distribution of topic z, $P(w_i|z) = \phi_{i|z}$ represents the probability distribution of topic z-specific word w_i, and $P(w_j|z) = \phi_{j|z}$ represents the probability distribution of topic z-specific word w_j.

By applying the chain rule on the joint probability of the whole data, the conditional probability can be obtained as follows:

$$P(z|z_{-b}, B, \alpha, \beta) \propto (n_z + \alpha)\frac{(n_{w_i|z} + \beta)(n_{w_j|z} + \beta)}{(\sum_w n_{w|z} + G\beta)^2} \tag{7}$$

where α and β are hyperparameters of Dirichlet distribution, n_z is the number of times of the biterm b assigned to the topic z, z_{-b} denotes the topic assignments for all biterms except b, $n_{w|z}$ is the number of times the word w is assigned to topic z. G represents the size of the vocabulary.

After determining the value of K, take $\alpha = 50/K$ and $\beta = 0.01$ according to experience. Topic distribution θ_z and topic-word distribution $\phi_{w|z}$ can be estimated as

$$\theta_z = \frac{n_z + \alpha}{|B| + K\alpha} \tag{8}$$

$$\phi_{w|z} = \frac{n_{w|z} + \beta}{\sum_w n_{w|z} + G\beta} \tag{9}$$

2) JS DIVERGENCE CALCULATION

After BTM modeling is completed, for each document, the first six specific words in $p(w|z)$ under the maximum probability topic in the document-topic distribution $p(z|d)$ are selected as the specific words of the document. The dimension based on the maximum retention of the document semantics and the complexity of the algorithm can be reduced [30]. Then, document d_i based on BTM topic modeling can be represented by a vector of posterior distribution of topics [14]:

$$d_{i\_BTM} = \{p(z_1|d_i), p(z_2|d_i), \ldots, p(z_K|d_i)\} \tag{10}$$

To calculate the similarity between two documents d_i and d_j, it is transformed into calculating the similarity between the two document-topic vectors d_{i_BTM} and d_{j_BTM}. JS divergence, a commonly used measure of text similarity, is used to calculate the text similarity here. The text similarity based on BTM topic modeling and JS divergence can be described as

$$\begin{aligned} &Dis(d_{i\_BTM}, d_{j\_BTM}) \\ &\quad = Dis_B(d_i, d_j) = Dis_{JS}(d_i, d_j) \\ &\quad = \frac{Dis_{KL}(d_i||\frac{d_i+d_j}{2}) + Dis_{KL}(d_j||\frac{d_i+d_j}{2})}{2} \end{aligned} \tag{11}$$

In Eq. (11), KL divergence is calculated as

$$\mathrm{Dis}_{KL}(p||q) = \sum_{h=1}^{6} p_h \ln \frac{p_h}{q_h} \tag{12}$$

where p and q are two probability distributions, p_h and q_h are the probability distributions of the first six specific words.

The text similarity measurement based on BTM topic modeling and JS divergence (BTM & JS-TSM) is shown in Algorithm 1.

Algorithm 1 BTM & JS-TSM

```
Input: Microblog short text set $D = \{d_1, d_2, \cdots, d_n\}$,
       hyperparameter $\alpha$ and $\beta$
Output: Text similarity based on topic $Dis_B(d_i, d_j)$
Determine the optimal number of topics $K$ according to perplexity
Initialize topic assignments randomly for all the biterms
for $d_i \in D$ do
    for $b \in B$ do
        Assign topics $z_b$ to each biterm according to Eq. (7)
        Update $n_z$, $n_{w_i|z}$ and $n_{w_j|z}$
    end for
end for
Calculate topic distributio $\theta_z$ and topic-word distribution $\phi_{w|z}$ according to Eq. (8) and Eq. (9)
Select specific words for each document and then represent texts according to Eq. (10)
Caculate the text similarity based on topic $Dis_B(d_i, d_j)$ according to Eq. (11)
Output the text similarity based on topic $Dis_B(d_i, d_j)$
```

C. TEXT SIMILARITY MEASUREMENT BASED ON GLOVE WORD VECTOR MODELING

The process of text similarity measurement based on GloVe word vector modeling is divided into two parts. The first part uses GloVe to model the preprocessed microblog short text set. The second part uses IWMD to calculate text similarity.

1) GLOVE WORD VECTOR MODELING

Considering that the cooccurrence ratio of two words with similar semantics is often higher, GloVe uses the cooccurrence ratio of words rather than the probability to learn the semantic similarity between words.

Therefore, before training the word vector, GloVe needs to count the cooccurrence times of target word v_s and context word $\tilde{v}_t$ in the whole corpus according to the size of the context window to construct the word cooccurrence matrix X_{st}. In the original paper, the optimal values of vector_size and window_ size used in the model were compared, and it was concluded that when vector_size = 300 and window_size is between 6 and 10, the model can achieve the best results. The dataset used in the source code is in English, so this paper sets the parameters vector_size = 300 and window_size = 8 according to the particularity of Chinese microblog short texts.

2) WMD OF IMPROVED WORD WEIGHT CALCULATION

After GloVe modeling, IWMD is used to calculate the text similarity. The text similarity between d_i and d_j based on GloVe word vector modeling and IWMD can be described as

$$Dis_G(d_i, d_j) = Dis_{IWMD}(d_i, d_j) = \min_{T\geq 0} \sum_{s,t=1}^{G} T_{st} c(s, t) \tag{13}$$

T_{st} is a G-order weight flow matrix where $T_{ij} \geq 0$ denotes how much of word s in document d_i travels to t in d_j. To transform d_i entirely into d_j, the entire outgoing flow from word s should equal w_pc_s, *i.e.*, $\sum_t T_{st} = w\_pc_s$, and w_pc_s is calculated according to Eq. (1). Furthermore, the amount of incoming flow to word t must match w_pc_t, *i.e.*,$\sum_s T_{st} = w\_pc_t$. The distance between the two documents is the minimum (weighted) cumulative cost required to move all words from d_i to d_j, *i.e.*,$\sum_{s,t} T_{st} c(s, t)$.

Formally, the minimum cumulative cost of moving d_i to d_j given the constrants is provided by the following linear program:

$$\min_{T\geq 0} \sum_{s,t=1}^{G} T_{st} c(s, t)$$

$$\text{subject to: } \sum_{t=1}^{G} T_{st} = w\_pc_s, \quad \forall s \in \{1, \ldots, G\} \tag{14}$$

$$\sum_{s=1}^{G} T_{st} = w\_pc_t, \quad \forall t \in \{1, \ldots, G\} \tag{15}$$

In Eq. (13), $c(s, t)$ is the word travel cost, which denotes the travel cost of word s in d_i travels to t in d_j, it can be calculated as

$$\mathrm{c}(s, t) = ||v_s - v_t||_2 \tag{16}$$

where v_s and v_t are GloVe word vectors of s and t.

The text similarity measurement based on GloVe word vector modeling and IWMD (GloVe & IWMD-TSM) is shown in Algorithm 2.

D. SHORT TEXT CLUSTERING BASED ON BTM AND GLOVE SIMILARITY LINEAR FUSIONXT

K-means is a clustering technology based on centroid, which defines the centroid of the cluster as the mean value of points in the cluster. The main principle of the algorithm is that K texts are randomly selected as the initial centers of K clusters in dataset D, the distances between other texts and each cluster center are calculated, and then texts are assigned to the most similar cluster. After that, the cluster center of each cluster is updated according to the calculation. Finally, the algorithm iterates repeatedly until the clustering criterion function converges.

Algorithm 2 GloVe & IWMD-TSM

```
Input: Microblog short text set $D = \{d_1, d_2, \cdots, d_n\}$,
       vector_size = 300, window_size = 8
Output: Text similarity based on word vector
       $Dis_G(d_i, d_j)$
Construct word co-occurrence matrix of microblog short
  text set $X_{st}$
Obtain word vector set $V = \{v_1, v_2, \cdots, v_G\}$ based on
  $X_{st}$ and GloVe modeling
for $s = 1$ to $G$ do
    Judge whether $s$ is the title word or the body word
      according to $l_s < 10$
    Update $c_{s\_title}$ and $c_{s\_body}$
    Specify Eq. (1) as the weight calculation formula and
    calculate the weight transform cost $w\_pc_s$
    Calculate weight transform matrix $T_{st}$ according to
      $\sum_t T_{st} = w\_pc_s$
end for
for $v_s, v_t \in V$ do
    Calculate word travel cost c$(s, t)$ according to
      Eq. (16)
end for
Caculate the text similarity based on word vector
  $Dis_G(d_i, d_j)$ according to Eq. (13)
Output the text similarity based on word vector
  $Dis_G(d_i, d_j)$
```

The cluster center c_e can be updated as

$$c_e = \frac{1}{n} \sum_{d_i \in C_e} d_i, \quad e = 1, 2, \cdots, K \tag{17}$$

where n is the total number of texts in the dataset, d_i is the i-th document, C_e represents the clustering set, and K represents the number of clusters.

Additionally, the clustering criterion function corresponding to the fusion similarity-based distance can be described as

$$\begin{aligned} E &= \sum_{e=1}^{K} \sum_{d_i \in C_e} Dis(d_i, c_e)^2 \\ &= \sum_{e=1}^{K} \sum_{d_i \in C_e} [\lambda \cdot Dis_B(d_i, c_e) + (1-\lambda) \cdot Dis_G(d_i, c_e)]^2 \end{aligned} \tag{18}$$

BG & SLF-Kmeans is shown in Algorithm 3.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL ENVIRONMENT AND DATASET

All the experiments are carried out on a PC, and the CPU is an Intel(R). The BTM and GloVe algorithms need to run in Linux environment. The processor is X64-based. The data acquisition software is Octopus V7.6.4. The experiments are compiled on Spyder of Anaconda3-5.2.0 with Python3.6.

This paper focuses on the hot topic discovery of the Sina microblog and selects the microblogs released by official news media and Sina "big V" as the dataset. It is generally agreed that the microblogs published by official news media and Sina "big V" are generally more influential and more likely to contain hot topics. Studying these microblogs can reduce the impact of nonhot topic microblog data on the experiment. Octopus is used to capture microblogs published by official news media and Sina "big V" from March 15, 2019 to March 19, 2019. After text preprocessing, 10,000 microblogs are retained as the dataset, of which 70% are retained as the training set, and 30% are retained as the test set. The date-quantity distribution of the dataset is shown in Table 1. The experimental data (part) is shown in Table 2.

Algorithm 3 BG & SLF-Kmeans

```
Input: Microblog short text set $D = \{d_1, d_2, \cdots, d_n\}$,
       the optimal number of topics $K$, the optimal
       fusion coefficient $\lambda$
Output: Specific word sets of $K$ topics
Randomly select $K$ short texts from dataset $D$ as the
  initial cluster centers $c_e$, $e = 1, 2, \cdots, K$
Specify Eq. (4) as the distance function
Repeat
    Calculate the distance between each short text $d_i$ and
      cluster centers $Dis(d_i, c_e)$,then assign each short text
      to the most similar cluster
    Update cluster centers $c_e$ according to Eq. (17)
Until Eq. (18) converges or cluster centers no longer
  change
Output the specific word sets of $K$ topics
```

TABLE 1. Date-quantity distribution of the dataset.

Date	Quantity of microblogs(pcs)
March 15, 2019	2,135
March 16, 2019	2,054
March 17, 2019	1,936
March 18, 2019	1,882
March 19, 2019	1,993

Each microblog is manually labeled with its topic label. According to statistics, the dataset has 12 topics in total, and the number of texts contained in each topic is sorted as shown in Table 3.

There are many topics on the Sina microblog every day, but only the Top-50 can be listed on the hot search list. That is, regardless of whether the hot topic of the day is more than 50 or less than 50, the Sina microblog defaults to the Top-50 as hot topics. The training set is used for multiple tests, and the number of texts corresponding to each topic is counted. The Top-6 topics are selected as hot topics, which are the most consistent with the ranking results in Table 3 and most in line with the actual situation in the dataset. Therefore, this paper assumes that

TABLE 2. Experimental data (part).

315 party # Medical garbage, hot bars, harassing phone calls, personal information disclosure, native eggs, license hanging, health products problems, after-sales traps for home appliances, electronic cigarettes, UnionPay quick pass, and high interest network loans are truly shocking.
A primary school teacher was sentenced to three years for molesting a 5-year-old girl, and five years later after the prosecution protested # The 5-year-old girl was sexually assaulted by a 56-year-old teacher, and the offender's reprimand was questioned.
UnionPay apologizes for the risk of brush stealing in quick pass and will further optimize the compensation mechanism # Consumers were warned of the risk of brush stealing in quick pass in the 315 party.
Reversal? Investigation results of Chengdu No.7 Experimental School: some people suspected to make false food photos # The photos of moldy and deteriorated food materials in the canteen of Chengdu No.7 experimental school may be artificial, and the suspected criminal clues are transferred to the public security organ.

TABLE 3. Topic-text quantity distribution.

Topic Label	Topic	Quantity of microblogs(pcs)
1	315 party	1,762
2	Li Shengli incident	1,508
3	A doctor's certificate is asked for when saving people on the high-speed rail	1,239
4	Chengdu No.7 Middle School food event	1,156
5	UnionPay apologizes	1,063
6	A teacher molests 5-year-old girl	1,041
7	Juvenile disobedience and mother killing	634
8	A Yunnan guy grabs the steering wheel to save 19 people	587
9	A girl is subjected to campus violence	316
10	Female college students selling eggs to repay online loans	288
11	Traffic police of punishment department collect money by eliminating points	234
12	BMW reduces recommended retail price	172

the Top-6 topics are hot topics. According to Table 3, the hot topics in the dataset are "315 party," "Li Shengli incident," "A doctor's certificate is asked for when saving people on the high-speed rail," "Chengdu No.7 Middle School food event," "UnionPay apologizes," and "A teacher molests 5-year-old girl."

B. EVALUATION INDEX OF CLUSTERING

In this experiment, Purity [31], F1-meausre [32] and Normalized Mutual information (NMI) [33] are used to analyze the results of clustering. F1-meausre depends on Precision (P) and Recall (R).

Purity represents the proportion of the number of correctly clustered texts to the total number of texts. It can roughly evaluate the clustering algorithm as a whole and can be calculated as

$$Purity(\Omega, D) = \frac{1}{n}\sum_{K}\max_{j}|C_k \cap d_j| \tag{19}$$

where $\Omega = \{C_1, C_2, \cdots, C_k\}$ represents the clustering set, C_k is the k-th clustering set, $D = \{d_1, d_2, \cdots, d_j\}$ represents the short text set, d_j is the j-th text, n is the total number of microblog short text set, K is the number of clusters, and $\max_{j}|C_k \cap d_j|$ represents the maximum number of documents in each cluster.

To further evaluate the clustering effect of a given cluster, the P, R, and F1-meausre of each cluster are calculated in this paper. P represents the proportion of the number of correctly clustered short texts to the total number of short texts in a given cluster, the larger P is, the better the cohesion of the clustering. R represents the proportion of short texts with the same topic clustered to the same cluster, the larger R is, the higher recognition rate of the algorithm. F1-measure is the harmonic mean of the two, only when both P and R are high is the clustering effect better. P, R, and F1-measure can be calculated as

$$P(i,j) = \frac{N_{ij}}{N_i} \tag{20}$$

$$R(i,j) = \frac{N_{ij}}{N_j} \tag{21}$$

$$F1 - \text{measure}(i,j) = \frac{2 \times P(i,j) \times R(i,j)}{P(i,j) + R(i,j)} \tag{22}$$

where $P(i,j)$, $R(i,j)$, and $F1 - \text{measure}(i,j)$ represent precision, recall, and F1-measure of topic i in topic j, N_{ij} is the number of texts belonging to topic i in the original dataset but to topic j in the clustering results, N_i and N_j are the number of short texts belonging to topic i and topic j.

When determining the optimal fusion coefficient λ, F1-measure of the final clustering results can be calculated as

$$F1 - \text{measure} = \frac{1}{n}\sum_{i=1}^{K}\max_{j} F1 - \text{measure}(i,j) \tag{23}$$

NMI is used to measure the similarity between the clustering results and the manually labeled results of the original dataset. The value is between 0 and 1. The closer the value is to 1, the more similar they are, that is, the more accurate the clustering results are. NMI can be calculated as

$$NMI = \frac{\sum_{i=1}^{K}\sum_{j=1}^{K} n_{ij}\log(\frac{n \cdot n_{ij}}{n_i n_j})}{\sqrt{(\sum_{i=1}^{K} n_i \log(\frac{n_i}{n}))(\sum_{j=1}^{K} n_j \log(\frac{n_j}{n})}} \tag{24}$$

where n is the total number of texts, K is the number of clusters, n_i and n_j are the number of texts belonging to topic i and j, n_{ij} denote the number of documents that are in topic i as well as in topic j.

C. SELECTION OF THE OPTIMAL NUMBER OF TOPICS

According to section III-B, the optimal number of topics can be determined by calculating perplexity. Repeat the experiment for 10 times and take the average value of 10 experimental results as perplexity corresponding to different K. The experimental results are shown in Fig. 3.

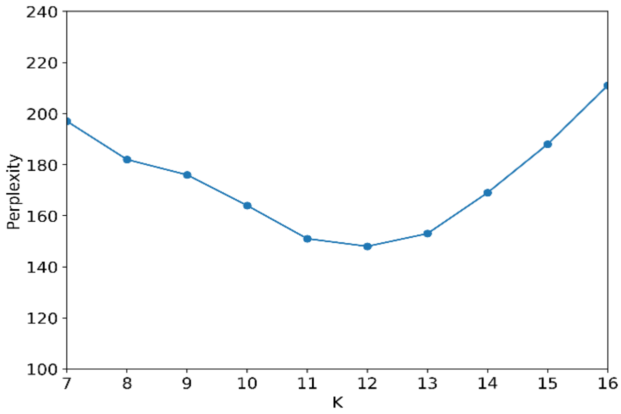

FIGURE 3. Perplexity of BTM corresponding to different K.

It can be seen from Fig. 3 that BTM has the least perplexity when $K = 12$. This indicates that the modeling effect is the best at this time, so the optimal number of topics is $K = 12$.

D. DETERMINATION OF THE OPTIMAL FUSION COEFFICIENT

In this section, $\lambda = 0.1, 0.2, \ldots\ldots, 0.9$, and the optimal fusion coefficient λ can be determined by calculating the F1-measure of the final clustering results. BG & SLF-Kmeans is repeatedly run 10 times, and the average value of the 10 experimental results is taken as the F1-measure of the final clustering results corresponding to different λ. The experimental results are shown in Fig. 4.

As shown in Fig. 4, when $\lambda = 0.6$, the F1-measure of the final clustering results is the highest. This indicates that the clustering effect of each topic is the best, so the optimal fusion coefficient $\lambda = 0.6$.

E. COMPARISON WITH OTHER HOT TOPIC DISCOVERY ALGORITHMS

According to the previous experimental results, the parameter is set to $K = 12, \lambda = 0.6$. In this paper, four algorithms are set to stop running when the cluster centers no longer change, and then the final clustering results can be obtained. The specific word sets of six hot topics obtained by running BG & SLF-Kmeans are shown in Table 4.

As shown in Table 4, six microblog hot topics are included in this dataset: “315 party,” “Li Shengli incident,” “A doctor’s certificate is asked for when saving people on the high-speed rail,” “Chengdu No.7 Middle School food event,” “UnionPay apologizes,” and “A teacher molests 5-year-old girl.” It can be seen that the hot topics obtained by BG & SLF-Kmeans are consistent with the results of manual labelling, which proves the effectiveness of this method.

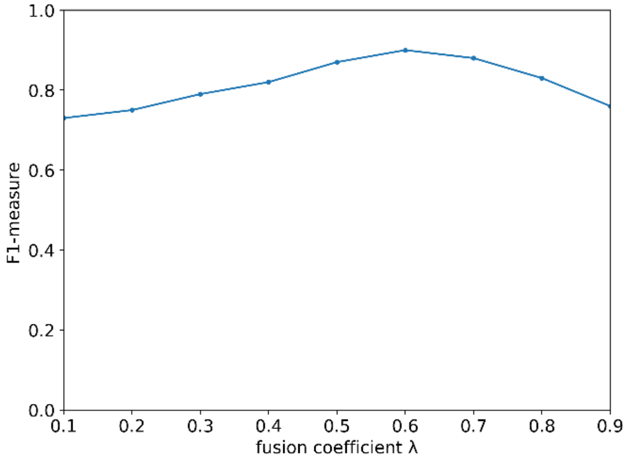

FIGURE 4. F1-measure corresponding to different λ.

TABLE 4. Specific word sets of six hot topics.

Topic Label	Specific Word Sets
1	315 party, brush stealing, electronic cigarettes, high interest network loans, after-sales traps for home appliances, medical garbage
2	Li Shengli, night club, sex entertainment, drug abuse,
	Zheng Junying, tax evasion
3	Save people on high-speed rail, ask for doctor's certificate, take photos for record, whole course video recording, nanning passenger section, apologize
4	Chengdu No.7 Experimental School, food, moldy, blood in the stool, false photos, artificial
5	UnionPay, quick pass, hidden danger of stealing, apologize, small amount secret free payment, full coverage
6	Teacher molest girls, sentence of second instance, heavier punishment, commuted for five years, question the change of sentence, light punishment

Qualitatively speaking, hot topics can also be extracted from the specific word sets obtained by the other three algorithms. However, compared with the specific word sets of the clustering results of the four algorithms, the discrimination is not high, so it cannot be explained which algorithm performs better. Therefore, comparison from a quantitative perspective is focused on in this paper. To verify the advantages of BG & SLF-Kmeans in clustering accuracy, it is compared with TF-IDF & K-means, BTM & K-means, and BTF & SLF-Kmeans in this paper. According to BTF & SLF-Kmeans, the optimal fusion coefficient $\lambda = 0.7$, so the optimal parameters of each algorithm are used in experiments. In this paper, the original dataset are labeled manually. The results of manual annotation are compared with the results of four algorithms, so that P, R, and F1-measure corresponding to each hot topic can be calculated. Each algorithm is repeated 10 times, and the average value is taken as the final result.

The Purity comparison of the clustering results is shown in Fig. 5.

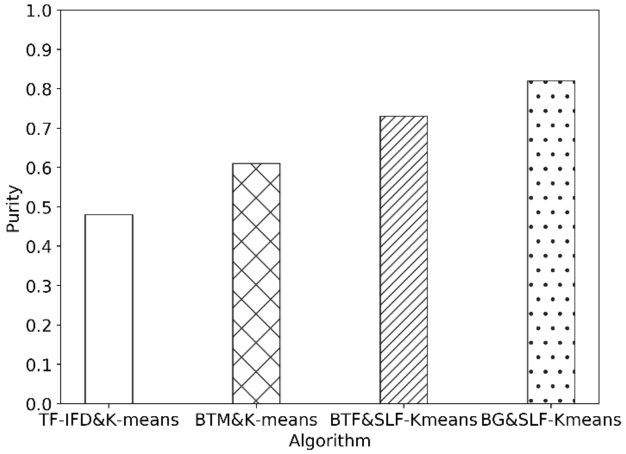

FIGURE 5. Purity comparison of clustering results.

From Fig. 5, the purity of the four algorithms gradually increases, and the purity of BG & SLF-Kmeans is the highest. Generally, compared with the other three algorithms, BG & SLF-Kmeans correctly clustered the most number of short texts, preliminarily verifying the effectiveness of this algorithm.

To further evaluate the clustering effect of each topic, P, R, and F1-measure corresponding to six topics are calculated. The comparison of P, R, and F1-measure of different topics is shown in Figs. 6-8.

From Figs. 6-8, the P, R, and F1-measure of BG & SLF-Kmeans are higher than those of the other three algorithms regardless of which topic. This further verifies the accuracy of this algorithm in clustering accuracy. The clustering effect of topic 1 and 5 are slightly worse, mainly because “315 party” and “UnionPay apologizes” have a sequential connection or even repeated content. Therefore, the semantic information is not clear when modeling and clustering errors occur. In these two topics, BG & SLF-Kmeans obtains a better clustering effect than other algorithms. It shows that positional contribution-based weight does improve the differentiation of words and further improve the accuracy of the similarity calculation.

To more accurately compare the clustering accuracy of the four algorithms, the F1-measure and NMI comparison of the final clustering results are further calculated. The result is shown in Fig. 9.

As is shown in Fig. 9, F1-measure and NMI of the final clustering results obtained by these four algorithms increases gradually, and the F1-measure and NMI of BG & SLF-Kmeans are the highest. It shows that both P and R of BG & SLF-Kmeans reach a good level. It also shows that the

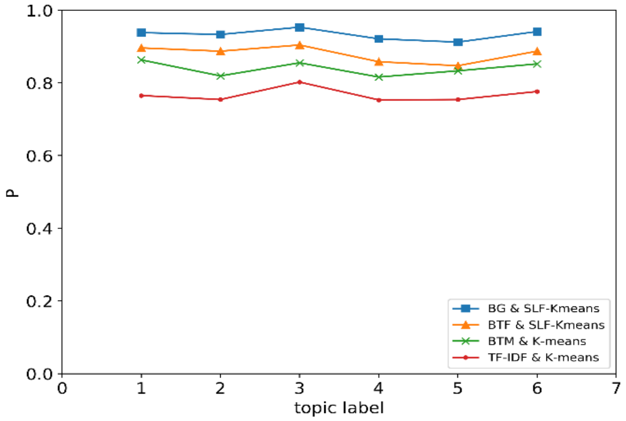

FIGURE 6. Comparison of P of different topics.

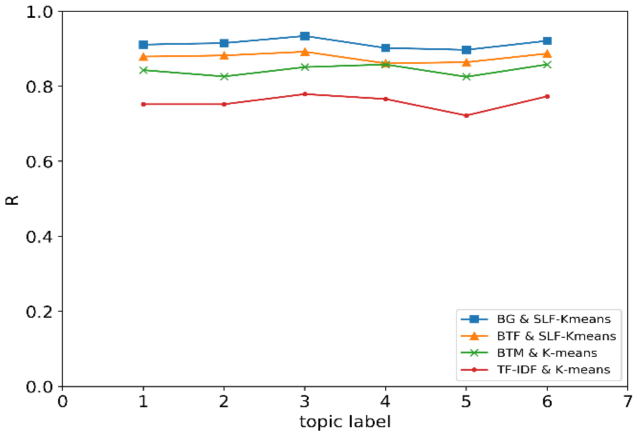

FIGURE 7. Comparison of R of different topics.

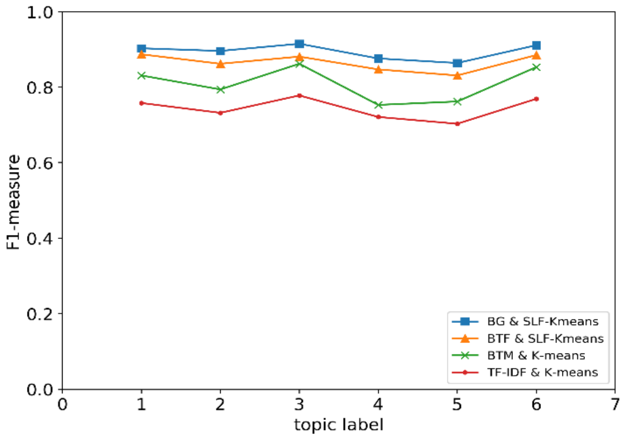

FIGURE 8. Comparison of F1-measure of different topics.

clustering result of BG & SLF-Kmeans is the most similar to the actual result of manual annotation. Overall, NMI can better reflect the advantages of BG & SLF-Kmeans in clustering accuracy. This proves again that BG & SLF-Kmeans

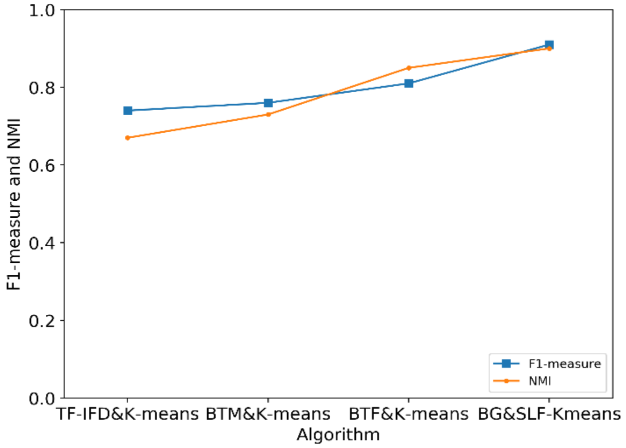

FIGURE 9. Comparison of F1-measure and NMI.

improves the clustering accuracy of microblog short texts and the quality of hot topic discovery.

V. CONCLUSION

This paper has studied the problem of how to accurately obtain hot topics from a large number of microblog short texts. A short text clustering algorithm based on BTM and GloVe similarity linear fusion (BG & SLF-Kmeans) was proposed. Due to the short length and special stylistic characteristics of news microblog texts, the definition of title words and body words was proposed. Different positional contribution of words were set up. To reflect the differences of words more accurately, the positional contribution-based weight was proposed. When calculating the similarity with distance function of K-means, the accuracy is not high, leading to inaccurate discovery of hot topics. Therefore, the advantages of BTM and GloVe in dealing with short texts are combined. Then, fusion similarity-based distance was proposed. In terms of Purity, F1-measure, and NMI, the experimental results show that BG & SLF-Kmeans achieved higher clustering accuracy than the other three algorithms. This proves that BG & SLF-Kmeans effectively improved the accuracy of microblog hot topic discovery.

REFERENCES

[1] L.-L. Shi, L. Liu, Y. Wu, L. Jiang, and J. Hardy, "Event detection and user interest discovering in social media data streams," *IEEE Access*, vol. 5, pp. 20953–20964, 2017.

[2] X. Geng, Y. Zhang, Y. Jiao, and Y. Mei, "A novel hybrid clustering algorithm for topic detection on chinese microblogging," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 2, pp. 289–300, Apr. 2019.

[3] J. Zhou, Y. Lu, H.-N. Dai, H. Wang, and H. Xiao, "Sentiment analysis of chinese microblog based on stacked bidirectional LSTM," *IEEE Access*, vol. 7, pp. 38856–38866, 2019.

[4] Y. Zheng, Z. Meng, and C. Xu, "A short-text oriented clustering method for hot topics extraction," *Int. J. Soft. Eng. Knowl. Eng.*, vol. 25, no. 3, pp. 453–471, Apr. 2015.

[5] D. Yan, E. Hua, and B. Hu, "An improved single-pass algorithm for chinese microblog topic detection and tracking," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, San Francisco, CA, USA, Jun. 2016, pp. 251–258.

[6] N. Yu, "A visualized pattern discovery model for text mining based on TF-IDF weight method," in *Proc. 10th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, Hangzhou, China, Aug. 2018, pp. 183–186.

[7] Y. Sun, H. Ma, M. Jia, and P. Wang, "An efficient microblog hot topic detection algorithm based on two stage clustering," in *Intelligent Information Processing VII*. Berlin, Germany: Springer, 2016.

[8] N. Zhou, P. Du, X. Jin, Y. Liu, and X. Cheng, "ET-TAG: A tag generation model for the sub-topics of public opinion events," *Chin. J. Comput.*, vol. 41, no. 7, pp. 1490–1503, 2018.

[9] Y. Lu, Q. Mei, and C. Zhai, "Investigating task performance of probabilistic topic models: An empirical study of PLSA and LDA," *Inf Retr.*, vol. 14, no. 2, pp. 178–203, Apr. 2011.

[10] Y. Chen, W. Li, W. Guo, and K. Guo, "Popular topic detection in chinese micro-blog based on the modified LDA model," in *Proc. 12th Web Inf. Syst. Appl. Conf. (WISA)*, Sep. 2015, pp. 37–42.

[11] M. Paul, "Cross-collection topic models: Automatically comparing and contrasting text," Dept. Comput. Sci., Urbana, Univ. Illinois Urbana-Champaign, Urbana, IL, USA, 2009.

[12] X. Chen, C. Ma, W. Wang, Y. Gao, and H. Wang, "Multi-source topic detection analysis based on improved ccLDA model," *Adv. Eng. Sci.*, vol. 50, no. 2, pp. 141–147, 2018.

[13] X. Chen, L. Luo, H. Wang, W. Wang, and Y. Gao, "Analysis and research on cross language topic discovery in Chinese and English," *Adv. Eng. Sci.*, vol. 49, no. 2, pp. 100–106, 2017.

[14] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, 2013, pp. 1445–1456.

[15] J. Feng and Y. Fang, "Research on hot topic discovery technology of micro-blog based on biterm topic model," in *Proc. Int. Conf. Geo-Inform. Resour. Manage. Sustain. Ecosyst.* Singapore: Springer, 2016, pp. 234–244.

[16] W. Li, Y. Feng, D. Li, and Z. Yu, "Micro-blog topic detection method based on BTM topic model and K-means clustering algorithm," *Autom. Control Comput. Sci.*, vol. 50, no. 4, pp. 271–277, 2016.

[17] Z. Li, J. Du, W. Cui, and P. Zhu, "User interaction based bursty topic model for emergency detection," in *Proc. Chin. Intell. Syst. Conf.* Singapore: Springer, 2019, pp. 11–21.

[18] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A probabilistic model for bursty topic discovery in microblogs," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 353–359.

[19] S. Wang, Y. Peng, and J. Wang, "Research of the text clustering based on LDA using in network public opinion analysis," *J. Shandong Univ. (Natural Sci.)*, vol. 49, no. 9, pp. 129–134, 2014.

[20] Y. Wang and Y. Hu, "Hotspot detection in microblog public opinion based on biterm topic model," *J. Intell.*, vol. 35, no. 11, pp. 119–124, 2016.

[21] Z. Zhu, J. Liang, D. Li, H. Yu, and G. Liu, "Hot topic detection based on a refined TF-IDF algorithm," *IEEE Access*, vol. 7, pp. 26996–27007, 2019.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[23] L. Guo, Y. Li, D. Mu, T. Yang, and Z. Li, "A LDA model based topic detection method," *J. Northwestern Polytech. Univ.*, vol. 34, no. 4, pp. 698–702, 2016.

[24] T. Lu, S. Hou, Z. Chen, L. Cui, and L. Zhang, "An intention-topic model based on verbs clustering and short texts topic mining," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Autonomic Secure Comput., Pervasive Intell. Comput.*, Liverpool, U.K., Oct. 2015, pp. 837–842.

[25] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[26] H. Imaduddin, W. Widyawan, and S. Fauziati, "Word embedding comparison for indonesian language sentiment analysis," in *Proc. Int. Conf. Artif. Intell. Inf. Technol. (ICAIIT)*, Yogyakarta, Indonesia, Mar. 2019, pp. 426–430.

[27] M. Kusner, Y. Sun, N. I. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 957–966.

[28] Z. Wang, X. Le, and Y. He, "Recognizing core topic sentences with improved textrank algorithm based on WMD semantic similarity," *Data Anal. Knowl. Discovery*, vol. 1, no. 4, pp. 1–8, 2017.

[29] W. Zhao, J. Chen, R. Perkins, Z. Liu, W. Ge, Y. Ding, and W. Zou, "A heuristic approach to determine an appropriate number of topics in topic modeling," *BMC Bioinf. BioMed. Central*, vol. 16, no. 13, p. S8, 2015.

[30] Y. Gao, Y. Xu, and Y. Li, "Pattern-based topics for document modelling in information filtering," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1629–1642, Jun. 2015.

[31] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Mach. Learn.*, vol. 55, no. 3, pp. 311–331, Jun. 2004.

[32] C. J. V. Rijsbergen, *Information Retrieval*. London, U.K.: Butterworth-Heinemann, 1979.

[33] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining partitionings," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Dec. 2002.

DI WU received the B.S. and M.S. degrees in computer application technology from the Hebei University of Engineering, and the Ph.D. degree in computer application technology from Yanshan University. She is currently an Associate Professor with the School of Software Engineering, Hebei University of Engineering, China. Her research interests include data mining, natural language processing, text clustering, and software security analysis.

MENGTIAN ZHANG is currently pursuing the master's degree with the Department of Information and Electronic Engineering, Hebei University of Engineering, China. Her research interests include data mining, natural language processing, text clustering, and recommendation systems.

CHAO SHEN received the Ph.D. degree in computer application technology from Shanghai University. He is currently a Lecturer with the School of Software Engineering, Hebei University of Engineering, China. His research interests include big data, cloud computing, data mining, and parallel computing.

ZHUYUN HUANG is currently pursuing the master's degree with the Department of Information and Electronic Engineering, Hebei University of Engineering, China. Her research interests include data mining, natural language processing, text clustering, and information retrieval.

MINGXING GU is currently pursuing the master's degree with the Department of Information and Electronic Engineering, Hebei University of Engineering, China. His research interests include data mining, recommendation systems, and parallel computing.