IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received October 19, 2020, accepted October 31, 2020, date of publication November 5, 2020, date of current version November 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3036043

A Novel Sentence Embedding Based Topic Detection Method for Microblogs

CONG WAN[1], SHAN JIANG[2], CONG WANG[1], YING YUAN[1], AND CUIRONG WANG[1]

[1]School of Computer and Communication Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China
[2]School of Computer and Communication Engineering, Northeastern University at Shenyang, Shenyang 110819, China

Corresponding author: Cong Wang (congw@neuq.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702089, in part by the Basic Scientific Research Operating Found of Central Universities under Grant N182304021, in part by the Scientific Research Plan for Institutions of Higher Learning of Hebei Province under Grant ZD2019306, and in part by the China Scholarship Council under Grant 201908130244.

ABSTRACT Topic detection is a difficult challenging task, especially when the exact number of topics is unknown. In this article, we present a novel topic detection approach based on neural computing to detect topics in a microblogging dataset. We use an unsupervised neural sentence embedding model to map blogs to an embedding space. The proposed model is a weighted power mean sentence embedding model in which weights are calculated by a targeted attention mechanism. The experimental results show that our embedding model performs better than baseline in sentence clustering. In addition, we propose a clustering algorithm, referred to as Relationship-Aware DBSCAN (RADBSCAN), to discover topics from a microblogging dataset in which the number of topics is automatically determined by the characteristics of the dataset. Moreover, to provide parameter insensibility, we use the forwarding relationship in the blogs as a bridge of two independent clusters. Finally, we validate the proposed method on a dataset from the Sina microblog. The results show that our approach can detect all topics successfully and can extract the keywords of each topic.

INDEX TERMS Topic detection, attention neural network, sentence clustering.

I. INTRODUCTION

In recent years, microblog platforms have become important vehicles for people to share opinions, explore new events, and disseminate information. In a microblog, various topics are discussed, advanced, and spread each day. Performing topic detection in microblogs is useful in many ways, such as for natural disaster detection [1], news recommendations [2], community detection [3] and political analyses [4]. Therefore, the detecting of topics from a large microblog dataset has become an important area of research interest.

Topic detection is one of the key tasks in sentiment analysis. A large number of studies have been conducted in this field. For example, Latent Dirichlet Allocation (LDA) [5] and various LDA-based models are widely used for topic detection. These models evaluate some topics, such as food, sports and the military, and then calculate the probability that a document belongs to a topic.

Neural networking is another popular technology that has been used for topic detection in recent years. They usually have an embedding layer to map documents to feature vectors as inputs to the network; then, various network structures, such as feed-forward neural networks [6], capsule neural networks [7], LSTMs [8] and Bi-LSTMs, can be used to output results. Topic detection is usually treated as a type of text classification task in neural networks [9], mostly as a type of supervised learning. However, we believe that unsupervised neural networks [6], [10] are more practical for topic detection.

However, topic detection of microblogs is quite different and difficult. First, there is no prior knowledge on the exact number of topics. For many previous topic models [11]–[13], the number of topics is an essential parameter. However, in practical application, the number of topics and what they are about are both unknown. For example, we do not know how many topics there were on Twitter last week until we perform a detailed data analysis. Second, microblogs are full of noise. For example, blogs about traffic are noises for natural disasters detection [1]; in contrast, a blog about weather

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

may be noise for traffic incident detection [14]. Therefore, previous studies obtained domain knowledge of topics to filter noise [11], [14]–[16]. Determining how to remove noise using a domain free method is still an open issue.

Feature extraction from short text is another key problem of topic detection models, which is especially useful for blogs in Chinese because they are typically short, dispersed, sparse and noisy [15], [17]. To resolve these problems, researchers usually adopt an embedding representation-based method [17]–[19] or a graph-based method [12], [13], [15] to extract topics. Furthermore, these two methods can be used together, similar to using GNN [20] for short text representation generation [21].

Graph-based methods can also be used for document relation structures [22], [23]. A microblog forwarding relationship is a type of document relation network that can provide much additional information [24]. However, this type of network has not been extensively used for topic detection.

So the objectives of our work is to detect topics from a microblog dataset and answer the following questions: (1) how many topics are there? (2) What are these topics about? (3) How to use microblog relation structure to improve performance?

In this article, we present a novel two step approach to resolve topic detection issues in microblogs. First, we design a sentence embedding neural model that maps a blog to an embedding space. These sentence embeddings are used for clustering, so we ensure that blogs with the same topic are mapped to nearby points. In our neural model, an attention mechanism is adopted to assign high weights to certain keywords to make important aspects clearer, and a power mean pooling layer is added to the model to enable the embedding space to store different kinds of information. Then, a clustering algorithm is proposed to detect clusters without knowing the K number of clusters, which we call a K-free algorithm. In this algorithm, we improve the DBSCAN [25] by introducing information about the network structures of the blogs. We consider two points to be in the same neighborhood when they have a connection in the microblog network no matter how far away they are. The experimental results show that our embedding method performs much better than other sentence embedding methods in clustering tasks and that our proposed clustering algorithm can accurately extract topics from a microblog dataset and performs better than comparison algorithms.

We summarize the main contributions of this article as follows:

1) We propose a novel neural model (PANM) to learn the microblog sentence embedding. In addition, PANM can extract keywords of the blog using attention using attention mechanism.

2) We make some specific improvement for DBSCAN and propose a novel clustering algorithm RADBSCAN.

3) We evaluate PANM an RADBSCAN against the state-of-the-art methods for topic detection. Experimental results demonstrate the superiority of our approaches.

The structure of this article is as follows. In Section II we review the current literature in this research area. In Section III we present the detail of our methods, including PANM and RADBSCAN. In Section IV we present our results with a discussion. We finally conclude this article in Section V.

II. RELATED WORKS

A. TOPIC DETECTION

In previous decades, a large amount of work has been performed on topic detection. For classical algorithms, the two main types of methods are text clustering based algorithms and model based algorithms. LDA is a relatively effective topic model based on probabilities. The main problem in LDA models is sparsity. A solution to this problem is to aggregate multiple short texts into one long text with the assistance of external information. An alternative solution is to extract information from a chosen long text, a priori, in some form first; but it is difficult to locate a suitable long text [26]. In recent years, a number of topic models based on LDA have been proposed. For example, Huang *et al.* [11] embedded an emotional layer in an LDA model, analyzed emojis and themes hidden in Weibo and proposed a multimodal joint emotional theme model. Amoualian *et al.* [5] presented an LDA-based model that generated topically coherent segments within documents by jointly segmenting documents and assigning topics to their words. Some scholars have used other methods to replace the above topic models and have also achieved good results. Gallagher *et al.* [27] proposed Topic Modeling with Minimal Domain Knowledge by Correlation Explanation to produce rich topics by optimizing the correlation explanation framework of sparse binary data. Yang *et al.* [23] made use of the rich link structure of the document network to improve topic coherence. Hida *et al.* [28] proposed a dynamic and static model, considering both the dynamic structure of temporal topic evolution and the static structure of each topic hierarchy. Xu *et al.* [18] proposed a simplified topic model (STM) that can infer the potential topic distribution of parameters as topic-level representations. Gurcan and Cagiltay [29] combined LDA with domain knowledge to detect most popular skills for software engineers.

For short texts, such as Weibo data, LDA models are not suitable for accurately learning Weibo text representations due to the noise, sparseness and lack of context of Weibo data. New clustering strategies for short texts have been designed. For example, Ni *et al.* [30] proposed a new clustering strategy, TermCut, to cluster short text fragments. The collection of short text fragments is modeled as a graph, and TermCut is applied to recursively select core terms and divide the graph equally. BTM [31] uses biterm for modeling short texts. BTM uses a window slider in the document and then uses the two words in the window as co-occurring word pairs. The model differs from LDA in that, after determining the topic distribution and word distribution, the co-occurring word pair is subsequently captured. BTM has successfully solved the LDA problem for short texts and has also solved the sparseness problem. Yan *et al.* [17] improved the traditional single-pass clustering algorithm and extracted the center vectors of multiple microclusters formed by the initial clustering to use for cluster merging. GPU-DMM [19] uses word embedding to enrich short texts. For a given short text, the GPU-DMM model first extracts a topic based on a conditional probability, selects the most relevant words in this topic, and uses the GPU model to expand the semantically related words of the selected words. Jia *et al.* proposed WordCom [13], which creates concept vectors by identifying semantic word communities in a weighted word co-occurrence network. TRTD [12] uses the closeness and importance of terms to discover closely related topic representative term groups. TRTD does not need to learn short text representations.

However, the problem with most of the above methods is that prior knowledge is required, such as domain knowledge or the number of topics. GSDMM [32] applied the Dirichlet Polynomial Mixture Model (DMM) to short text clustering for topic detection. KSIHK [33] built a three layers clustering framework to detect topics from microblog. GSDMM and KSIHK can work without the number of topics, but they are initial parameter sensitive. According to the review of topic detection, we decided that the objective of our work is to detect topics without prior knowledge.

B. SENTENCE EMBEDDING

In recent years, many researchers have studied various methods for learning sentence representations. Discovering ways to use word embeddings to express a sentence has become an important research focus. In fact, sentence embeddings are widely used in tasks such as information retrieval, sentence classification, and topic detection. Many literature [17]–[19], [34]–[36] describes the use of embedding technology for topic detection. Previous studies [17], [34], [35] and [19] use the word2vec technology proposed by Google, and [18] uses GCN-based Encoder technology. Curiskis *et al.* [36] benchmarked four embedding methods, including word2vec and doc2vec, in Twitter and Reddit dataset to detect topics. Zhang *et al.* [37] used LSTM to learn embedding of short text in topic detection task.

There have been new developments of sentence embedding in recent years. Reference [38] combines CNN and RNN networks to represent sentence and text classifications. Reference [39] combines CNN and LSTM to achieve single sentence representations. Arora *et al.* [40] claimed to have found a simple but tough-to-beat baseline model. Each sentence is expressed as a weighted average of the embedding of the contained words; then, the sentences are put together to find the largest major axis; and finally, the largest major axis is removed from each sentence. Conneau *et al.* [41] use SNLI (Stanford Natural Language Inference Corpus) to train the model in advance and then transfer learning. In SNLI, each pair of sentences is marked as one of three categories: entailment, contradiction, and neutral. The selected model BiLSTM-Max works best and outperforms the other models. Logeswaran and Lee [42] proposed a simple and efficient framework named QT (Quick Thoughts) based on Skip-thought. The difference between QT and Skip-thought is that QT is directly output to a classifier after encoding and evaluates the adjacent sentence. The advantage of this process is that the speed of operation is greatly improved. However, most sentence embeddings do not consider the effect of short text clustering. For example, JC (Jaccard Coefficient) and NMI (Normalized Mutual Information) are rarely used as evaluation indicators of sentence embeddings. Learning sentence embedding is the first step of our method, the objective of this step is to design a method more suitable for clustering.

C. NEURAL NETWORK BASED TEXT CLASSIFICATION

Topic detection is a core task of text classification, so many researchers perform studies from a text classification perspective. In recent years, research on text classifications based on neural networks has become mainstream [9]. For example, Chen *et al.* [16] used deep learning methods to detect traffic topics from Weibo texts. They used convolutional neural network (CNN) models, long short-term memory (LSTM) models and their combination, LSTM-CNN models, to extract relevant Weibo content. Magdy and Elsayed [15] proposed an adaptive and completely unsupervised method to filter the content generated by Weibo users and track a wide range of dynamic topics.

For example, fastText [43] superimposes the words and n-gram vectors of an entire document to obtain the document vector and then uses the document vector to perform softmax multiclassifications. However, fastText and DAN (deep average network) [44] do not consider the word order because the input layer uses an addition and averaging operation. In regards to TextCNN models, Kim [45] proposed the basic structure of TextCNN, and Zhang and Wallace [46] performed an experimental comparison that controlled the TextCNN model variables. Recurrent neural networks (RNNs) are more commonly used in Natural Language Processing than other networks. Reference [47] designed three information sharing mechanisms based on RNNs; that is, multiple tasks were combined for training. Compared with CNNs, which ignore ‘‘structural information’’, the capsule layer in the capsule network [7] improves the expressive ability of the model, and its output is a vector containing information such as position, size, and angle. Zhao *et al.* [48] improved the capsule network and applied it to text classification, where capsules represent sentences or documents. There were also some attempts to add attention mechanisms. Reference [49]’s model applied two levels of attention mechanisms at the word and sentence levels; Lin *et al.* [50] used a self-attention mechanism to learn sentence embedding. The memory enhancement network combines a neural network with external memory, and the model can learn which text sequences are input and output. Transformers apply self-attention mechanisms to compute each word in parallel, overcoming the limitation of RNNs for increasing the computational cost of long sentences. TextRank [51] introduced graph theory into the task of text sequencing and used graphs to express syntactic and semantic relations. Yao *et al.* [21] used graph convolutional networks [20] for text classification and Text GCNs to learn predictive words and document embeddings. There are also some mixed methods. For example, C-LSTM [38] uses CNNs to extract advanced phrase representation sequences and then inputs them into the LSTM network to obtain sentence representations; [39] uses CNNs to learn sentence representations and uses gated RNNs to learn document representations that encode internal relationships between sentences.

The previous literatures show that neural network plays an important role in the topic detection framework, and the network structure needs to be designed according to the specific task background.

D. SOCIAL NETWORK STRUCTURE ANALYSIS

Social networks, such as Weibo, are where subject discovery is primarily applied. In social network analysis, network structure information is as important as semantic information [52], [53]. Forwarding network is one of the characteristics of microblog. Fu *et al.* [24] proposed a model to establish the relationship between topic and use forwarding behavior.

In recent years, many researchers have devoted themselves to detect topics from microblogs. Gu *et al.* [14] used Twitter to detect traffic. Compared with the data collected by traditional event detectors, Twitter data have a relatively high true positive rate and sampling rate, although the text is not standardized. Similarly, Chen *et al.* [16] uses Weibo data to detect traffic information. Unlike traditional text, Weibo data are multimodal. In a sentiment analysis based on microblog data, MJST [11] stated that it solved the problem endemic to most existing detection methods of treating microblog data as noise. Yan *et al.* [17] focused on the financial field and used Weibo for topic detection and tracking. In addition, Weibo also has a forwarding function, and this process brings about the rapid dissemination and diffusion of information. However, these methods all depend on domain knowledge.

III. METHOD

In this section, we introduce our approach in detail. First, a power-mean and attention-based neural model (PANM) is proposed to generate vectors from microblogs; this model is unsupervised. Second, a novel clustering algorithm, referred to as RADBSCAN, is proposed to detect topics.

A. NEURAL MODEL

The goal of our PANM model is to learn sentence embeddings. The structure of PANM is shown in Figure 1, and the meanings of the notations in this section are shown in Table 1. First, microblogs are segmented, and some feature words are selected as inputs to our model.

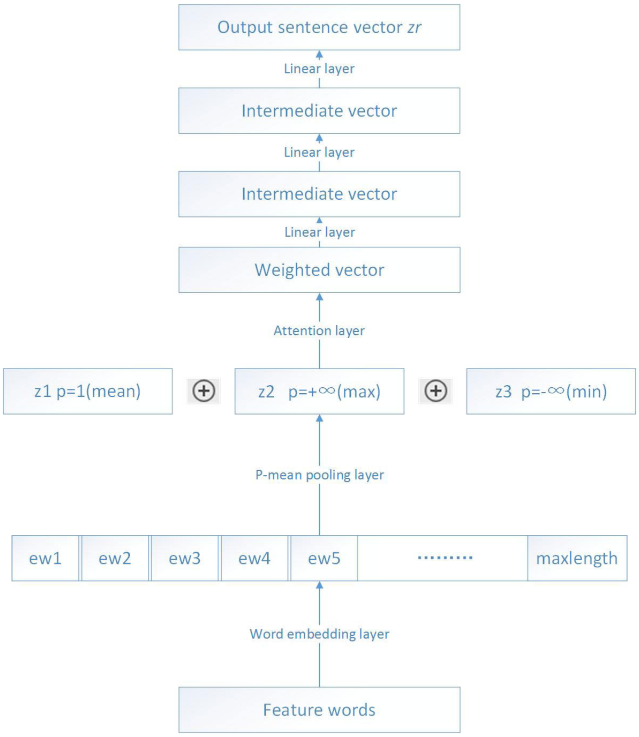


FIGURE 1. PANM model.

TABLE 1. Meanings of the notations.

Notation	Description
e_w	Word vector
R^d	A d-dimensional real number space
z	Sentences vector which has the same dimension as e_w, it will be used as input to PANM.
a_i	The weight of the word i.
p	A real number used to represent a power.
zr	The sentence embedding output by PANM.

We find that only nouns, adjectives and verbs can be used to extract topic information in Chinese, so we only keep these three types of words. Then, stop words, numbers, punctuations, URLs and useless words, such as "@someone", are removed. Table 2 shows an example of a microblog and its feature words. For English readers, we also provide the English version. Obviously, most of the feature words are consistent with the topic of the microblog. However, there is still an unexpected word, gas (i.e., 气 in Chinese). This is a segmentation error. Chinese word segmentation is very difficult, and errors are inevitable. In the rest of this section, we describe a neural model that can reduce the impact of these errors by automatically learning word weights.

Each feature word is associated with a feature vector $e_w \in R^d$, where d is the dimension of the word embedding. Then, we construct a sentence embedding z from the vector e_w by a power mean based method [54]. This method generalize the idea of generating sentence embeddings by averaging

TABLE 2. An example of feature words selection.

Original Chinese microblog	下次买手机依旧是魅族,但是体验过小米在线购物过程觉得很舒心(不是抢手机...),魅族的在线购物体验还有很多时候跟用户的互动比小米差太多了,小米接地气,魅族装格调却没装好,加油吧 @黄章
Microblog in English	Although shopping on the Xiaomi website is very comfortable, I will choose a Meizu mobile phone next time. Meizu website's user experience and interaction are much worse than Xiaomi. Xiaomi is down to earth,and Meizu pretends to be very stylish. Meizu should do better. @黄章
Feature words in Chinese	'买手机', '魅族', '体验', '小米', '购物', '过程', '用户', '气', '装', '格调'
Feature words in English	buy a mobile phone, Meizu, experience, Xiaomi, shopping, process, user, gas, pretend, style

* 黄章 is a microblog user, @黄章indicates that this microblog is related to him.

word embeddings, which is described in Equation (1), where p defines the mean type, such as the arithmetic mean ($p = 1$), the geometric mean ($p = 0$), or the harmonic mean ($p = -1$).

$$z = (\sum_{i=1}^{n} e_{wi}^{p})^{1/p} \tag{1}$$

In our work, the power mean method is weighted as in (2), so some additional important keywords could have a greater impact on the sentence position in the embedding space, which could make sentences with the same topic closer.

$$z = (\sum_{i=1}^{n} a_i e_{wi}^{p})^{1/p} and \sum_{i=1}^{n} a_i = 1 \tag{2}$$

Moreover, to obtain richer summary statistics of the sentence, a few different values of p are selected, and the sentence embeddings are concatenated together. Thus we obtain a new embedding z.

$$z = (\sum_{i=1}^{n} a_{1i} e_{wi}^{p_1})^{1/p_1} \oplus (\sum_{i=1}^{n} a_{2i} e_{wi}^{p_2})^{1/p_2} \oplus (\sum_{i=1}^{n} a_{3i} e_{wi}^{p_3})^{1/p_3} \tag{3}$$

The word weight a is computed by an attention-based method.

$$y = \frac{(\sum_{i=1}^{n} e_{wi}^{p})^{1/p}}{n} \tag{4}$$

$$d_i = e_{wi}^{T} \cdot M \cdot y \tag{5}$$

$$[a_i = \frac{e^{d_i}}{\sum_{i=1}^{n} e^{d_i}}] \tag{6}$$

Finally, we reconstruct the sentence embedding zr through three linear layers. This process is described in the following Equation (7), where $g(\cdot)$ is the activation function ReLU.

$$zr = g(g(g(z^T \cdot M_1) \cdot M_2) \cdot M_3) \tag{7}$$

Thus, the goal of training PANM is to learn the parameter matrices, M, M1, M2 and M3.

Having described our model structure, we now introduce our loss function L, which is based on the hingle loss function [6], [55]. For training, we use L to guarantee the following targets: 1) there is a high inner product between zr and z; 2) there are low inner products between zr and the negative samples. Our loss function is described in (8), where $s_i \in S$ is a negative sample.

$$L = \sum_{i=1}^{m} \max(0, 1 - z^T zr + zr^T s_i) \tag{8}$$

The negative samples are randomly selected from the microblog dataset. When we obtain a random blog, we associate its feature words to word embeddings as the first layer of PANM. Then, the negative sample s is generated by an unweighted power mean concatenation method in (9).

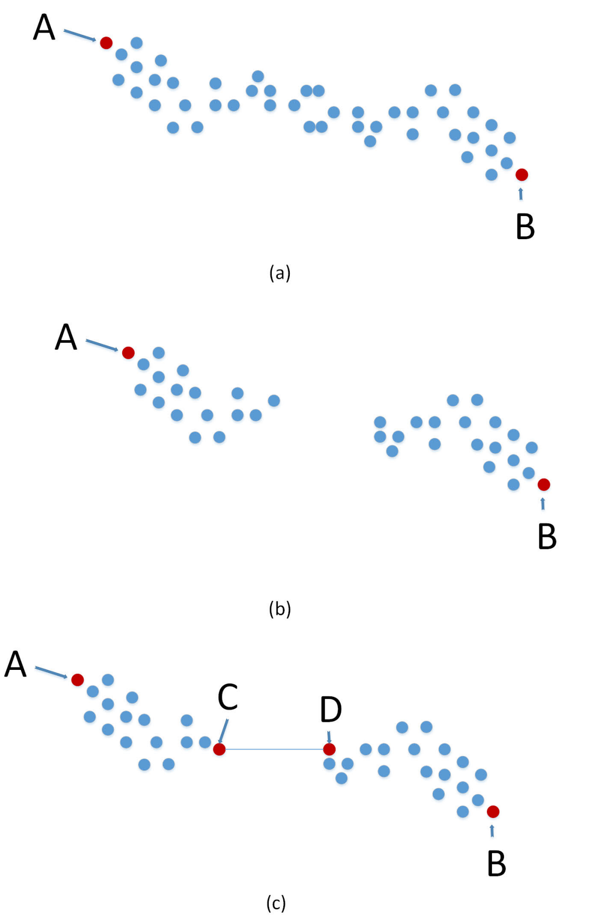
$$s = (\sum_{i=1}^{n} e_{wi}^{p_1})^{1/p_1} \oplus (\sum_{i=1}^{n} e_{wi}^{p_2})^{1/p_2} \oplus (\sum_{i=1}^{n} e_{wi}^{p_3})^{1/p_3} \tag{9}$$

B. RELATIONSHIP-AWARE DBSCAN (RADBSCAN)

In this subsection, we describe our algorithm, relationship-aware density based spatial clustering of applications with noise(RADBSCAN), which makes use of the forwarding relationship between blogs. The main idea of our algorithm is to expand the definition of density-reachable [25], which is the core concept of DBSCAN.

If there is a path from point A to point B, which is surrounded by points whose density exceeds a certain threshold, then point A and point B are density-reachable, as illustrated in Figure 2(a). In contrast, point A and point B are density-unreachable, as shown in Figure 2(b), because the path is broken. However, in our opinion, if there is an exact connection between point C and point D, then the connection can rebuild the path like a bridge. Therefore, A and B are still density-reachable in Figure 2(c). From the perspective of density based clustering, the density-reachable points are considered to be in the same cluster; thus, there is only one cluster in Figure 2(a) and (c) and are two clusters in Figure 2(b). Supposing C and D are blogs, the forwarding relationship between them can be this bridge. Forwarding is an important function of a microblog. Forwarding means that you can post one microblog on a website and refer it to another one. Generally speaking, the two microblogs focus on the same topic.

Next, we illustrate the forwarding relationship. There are three blogs in Table 3, blog 1 forwards to blog 3, and blog 3 forwards to blog 2. These three microblogs engaged in a discussion of which mobile phone brand to choose, but it is



(a)

(b)

(c)

FIGURE 2. The extension of the definition of density-reachable.

TABLE 3. Example of microblog forwarding.

	Original Chinese microblog	Microblog in English	Key words
Blog 1	你居然抛弃了魅族……	You abandoned Meizu.	Meizu
Blog 2	好吧，你们让我犯了选择综合征，努比亚确实也不错，还是等考完试再买吧	Well, you made it hard to choose. Nubia is also really good. I'll make a decision after my exam.	Nubia
Blog 3	认真看了大家的推荐，华为p6有点温婉了，就决定是小米3吧！谢谢大家么么哒	I will choose Xiaomi instead of Huawei due to everybody's recommendation. Huawei is not suitable for my style. Thank you very much!	Huawei; Xiaomi

* Meizu, Nubia, Huawei and Xiaomi are all mobile phone brands.

difficult for text clustering to put them in the same cluster without domain knowledge of mobile phones.

Based on the above discussion, the RADBSCAN algorithm is proposed. From the perspective of clustering algorithms, sentence vectors are regarded as points in an embedding space. Therefore, in the description of our proposed algorithms, the notion of a point represents a sentence.

```
Algorithm 1 RADBSCAN
Require: D, eps, MinPts
Ensure: clusters with different labels C
C = 0
for all point P in dataset D do
    if P is visited or P is NOISE then
        continue next point;
    end if
    [NeighborPts, RelatedPts] = regionQuery(P, eps)
    if sizeof(NeighborPts) < MinPts then
        mark P as NOISE
    else
        C = C+1
        mark P as visited
        NeighborPts= NeighborPts joined with RelatedPts
        expandCluster(P, NeighborPts, C, eps, MinPts)
    end if
end for
```

```
Algorithm 2 RegionQuery
Require: P, eps
NeighborPts= all points within P's eps-neighborhood
    (including P)
RelatedPts=all points having forwarding relationship
return [NeighborPts, RelatedPts]
```

In addition, there are important concepts in RADBSCAN, which are described as follows:

Definition 1: neighborhood-of-a-point (short for *eps*) is a threshold defining the range of neighbors. If the distance of two points is less than *eps*, then they are in the same neighborhood.

Definition 2: minimum-number-of-points (short for *MinPts*) is a threshold defining the minimum number of points in the same neighborhood.

Definition 3: Forwarding relationship set (represented by *FR*). p and q are two points obtained by mapping microblogs T1 and T2 to vector by PANM. If T1 forwards to T2, then $< p, q > \in FR$ and $< q, p > \in FR$.

Definition 4: Core point. A point p is a core point if: $|\{q | dist(p, q) \leq eps\}| + |\{q| < q, p > \in FR\}| \geq MinPts$.

Definition 5: Directly density-reachable. A point p is directly density-reachable from a point q if: 1) q is a core point; 2) $dist(p, q) \leq eps$ or $< q, p > \in FR$.

Definition 6: Density-reachable. A point p is density reachable from a point q if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_i+1 is directly density-reachable from p_i. This definition is consistent with that in DBSCAN. It can be concluded that the main differences between DBSCAN and our RADBSCAN are as follows:

1) We define the forwarding relationship set *FR*, which is new;

2) We redefine the core point and directly density-reachable.

TABLE 4. Dataset identifiers and statistics.

identifiers	Topics	Microblog number	Number of microblogs filtered	Average length	Vocabulary size
Dataset A	Meizu, Rocket, house prices, civil servants and smog	15000	12608	10	14603
Dataset B	Meizu, Rocket, house prices, civil servants	12000	10123	10	11857
Dataset C	Meizu, Rocket, house prices	9000	7362	9.2	8973

```
Algorithm 3 ExpandCluster
Require: P NeighborPts, C, eps, MinPts
Ensure: Cluster C with all of its members
add P to cluster C
for all point P' in NeighborPts do
    if P' is not visited then
        mark P' as visited
        [NeighborPts', RelatedPts']= regionQuery(P', eps)

        if sizeof(NeighborPts') >= MinPts then
            NeighborPts = NeighborPts joined with NeighborPts'
        end if
        NeighborPts= NeighborPts joined with RelatedPts'
    end if
    if P' is not yet member of any cluster then
        add P' to cluster C
    end if
end for
```

RADBSCAN is described by the following three algorithms: algorithm 1 is the main function and algorithm 2 and algorithm 3 are called by algorithm 1. In algorithm 1, *D* is the whole point dataset, and each point in *D* has a label and a state. The label indicates which cluster it belongs to, and none of the points have labels in the beginning. Each point in *D* belongs to one of the following three states: visited, noise or undefined. Visited points already have a cluster label; noise points are generally noise without a label, but there is still a chance of obtaining a label; undefined is the initial state.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed method including both the sentence embedding model PANM and the clustering algorithm RADBSCAN.

A. DATASET

Our evaluation is based on a real microblog dataset that was web-crawled from www.sina.com. The dataset can be download from https://ieee-dataport.org/documents/microblog-dataset.

There are 15000 blogs in the dataset, including five topics: cellphone, basketball, house prices, civil servants and smog. Meizu is a popular Chinese mobile phone brand, and Rocket is an NBA team. We use Jieba(https://pypi.org/project/jieba/) for Chinese word segmentations, and we filter the feature words according to the part of speech. Then, we use a stop word list published by Baidu and HIT for further filtration. Some sentences have no feature words left, and they are discarded.

To test the performance of our method for different topic numbers and microblog numbers, we randomly extract topics from the entire dataset and form several subsets. Their identifiers and statistics are presented in Table 4. The statistics for the topics are in Table 5.

TABLE 5. Topic statistics.

Topic	size	Average length	Vocabulary size
Meizu	2371	7.8	3170
Rocket	2254	8.2	4102
House prices	2737	11.5	4579
Civil servants	2761	11.9	5566
Smog	2485	10.2	5808

B. SENTENCE EMBEDDING PERFORMANCE EVALUATION

In this work, we propose a sentence embedding model, PANM, through which sentence vectors are generated; then, these vectors are used for short text clustering. Our model is trained on a PC with 8 GB of memory. The software environment is python3.6, pycharm, pytorch and numpy. The word embedding dimension is 100, the epoch is 10, the negative sample set size is 20, the optimizer is adam and the learning rate is 0.001.

To evaluate PANM, we choose the following sentence embedding methods as baselines and then performing clustering with K-mean and EM. The results of the clustering can be used to evaluate the model. The baselines are as follows:

1) POWER-MEAN [54]

This model is similar to ours in that it adopts a variety of pooling methods and joins them; however, there are obvious differences. We use the attention model to give weight to every word in a sentence so that the most important word has the highest weight.

2) TF-IDF-SVD

The feature of a sentence can be represented by a sparse vector of TF-IDF, and then, the vector is reduced by SVD.

3) SIF [40]

This is the most common baseline for sentence embedding.

TABLE 6. Experimental results for sentence embedding methods on Dataset A.

	K-mean					EM				
	NMI	JC	RI	FMI	Precision	NMI	JC	RI	FMI	Precision
PANM	**0.628**	0.391	**0.782**	**0.574**	**0.726**	**0.609**	**0.387**	**0.789**	**0.579**	**0.67**
Power-mean	0.619	**0.406**	0.785	0.532	0.759	0.601	0.365	0.741	0.538	0.669
TF-IDF-SVD	0.57	0.327	0.742	0.505	0.703	0.456	0.34	0.781	0.51	0.632
SIF	0.576	0.347	0.747	0.53	0.691	0.421	0.304	0.753	0.471	0.578
Simple Word Averaging	0.578	0.350	0.751	0.532	0.696	0.446	0.324	0.769	0.493	0.616
Keywords-averaging	0.624	0.402	0.726	0.517	0.714	0.427	0.277	0.645	0.465	0.558

4) SIMPLE WORD AVERAGING [56]

SWA is a simple but high performance method. It is the inspiration for many of the other methods.

5) KEYWORDS- AVERAGING

We can extract the three most important keywords according to the word weights from the three pooling methods (mean, max, min) in PANM. KA is a simple average of these three keywords' embeddings.

We choose metrics for the performance evaluations. Before we introduce these metrics, let us define some symbols. Given a set of N samples $S = \{o_1, o_2 \ldots o_N\}$, let $C = \{c_1, c_2 \ldots c_k\}$ be the true partition of the sample set and $\Omega = \{\omega_1, \omega_2 \ldots \omega_k\}$ be the calculated partition of the sample set.

1) NMI (NORMALIZED MUTUAL INFORMATION)

NMI is a measure of the similarity between two labels of the same sample set. NMI can be calculated as in (10). The values of NMI range from 0 to 1. The larger the value, the better the classification effect.

$$NMI = \frac{2\sum_{i=1}^{k}\sum_{j=1}^{k}\frac{|\omega_i \cap c_j|}{N}\log\frac{N|\omega_i \cap c_j|}{|\omega_i||c_j|}}{-\sum_{i=1}^{k}\frac{|\omega_i|}{N}\log\frac{|\omega_i|}{N} - \sum_{j=1}^{k}\frac{|c_j|}{N}\log\frac{|c_j|}{N}} \tag{10}$$

2) RI (RAND INDEX)

RI is used for measuring the similarity between two data clusterings. RI is similar to accuracy, but it is more widely used in measurements of clustering algorithms. The JC (Jaccard Coefficient) can be calculated as shown in (11), in which $a = |S^*|$ where $S^* = \{(o_i, o_j)|o_i, o_j \in \omega_l, o_i, o_j \in c_g\}$ and $b = |S^*|$ where $S^* = \{(o_i, o_j)|o_i \notin \omega_l, o_j \in \omega_l, o_i \notin c_g, o_j \in c_g\}$.

$$RI = \frac{a+b}{2N(N-1)} \tag{11}$$

3) JC (JACCARD COEFFICIENT)

The JC is a statistic used for gauging the similarity and diversity of sample sets. The JC can be calculated as shown in (12), in which $a = |S^*|$, where $S^* = \{(o_i, o_j)|o_i, o_j \in \omega_l, o_i \notin c_g, o_j \in c_g\}$.

$$JC = \frac{a}{a+b+c} \tag{12}$$

4) FMI (FOWLKES AND MALLOWS INDEX)

FMI is an external evaluation method, which is used to determine the similarity between two clusterings. FMI can be calculated as shown in (13).

$$FMI = \sqrt{\frac{a}{a+b}\cdot\frac{a}{a+c}} \tag{13}$$

5) PRECISION

Precision can be calculated as (14).

$$\Pr ecision = \sum_{i=1}^{k}\frac{|\omega_i|}{N}\max\{\frac{|c_j \cap \omega_i|}{|\omega_i|}, c_j \in C\} \tag{14}$$

Then, we cluster those sentence embeddings with K-means and EM algorithms. The results are as shown in Table 6, 7 and 8. It can be found that our method performs best in almost all metrics, especially the NMI, which proves that our method can extract the features of a microblog for Chinese text clustering better than other methods. For some results, Keywords-averaging (KA for short) and power-mean also performed well, which is in line with our expectations. These two methods are actually equivalent to two layers of PANM. KA has the same effect as the attention layer, keywords used by KA are extracted by the attention layer of PANM. And Power-mean has the same effect as the P-mean pooling layer. So PANM can take advantage of these methods and show good performance.

C. EXPERIMENTAL ANALYSIS OF RADBSCAN

To show the effectiveness of the proposed algorithm, performance evaluations based on the foregoing work are conducted. The goal of this work is to pick out topics worthy of further attention given a large amount of noise. DBSCAN is designed for this purpose; however, there are some disadvantages. In this section, we show that our RADBSCAN enhances DBSCAN and avoids its two disadvantages. First, DBSCAN is parameter sensitive. The value of *eps* has a strong influence on the clustering results. Second, high dimensions reduce the performance of DBSCAN.

TABLE 7. Experimental results for sentence embedding methods on Dataset B.

	K-mean					EM				
	NMI	JC	RI	FMI	Precision	NMI	JC	RI	FMI	Precision
PANM	**0.616**	**0.441**	**0.780**	0.616	**0.777**	**0.643**	**0.487**	**0.803**	**0.660**	**0.778**
Power-mean	0.609	0.417	0.777	0.563	0.770	0.633	0.406	0.772	0.656	0.772
TF-IDF-SVD	0.513	0.337	0.685	0.516	0.684	0.420	0.372	0.733	0.548	0.651
SIF	0.524	0.357	0.692	0.540	0.672	0.446	0.348	0.746	0.517	0.629
Simple Word Averaging	0.534	0.365	0.701	0.548	0.682	0.426	0.325	0.729	0.491	0.627
Keywords-averaging	0.607	0.437	0.772	**0.631**	0.776	0.425	0.299	0.625	0.480	0.612

TABLE 8. Experimental results for sentence embedding methods on Dataset C.

	K-mean					EM				
	NMI	JC	RI	FMI	Precision	NMI	JC	RI	FMI	Precision
PANM	**0.509**	**0.447**	**0.713**	**0.622**	**0.757**	**0.454**	**0.431**	**0.714**	**0.605**	**0.744**
Power-mean	0.485	0.423	0.709	0.768	0.717	0.421	0.344	0.690	0.606	0.730
TF-IDF-SVD	0.411	0.366	0.611	0.548	0.664	0.421	0.43	0.712	0.61	0.733
SIF	0.501	0.403	0.700	0.612	0.704	0.322	0.345	0.656	0.514	0.741
Simple Word Averaging	0.506	0.408	0.704	0.616	0.709	0.324	0.332	0.641	0.5	0.647
Keywords-averaging	0.497	0.407	0.713	0.611	0.737	0.277	0.326	0.590	0.499	0.616

Figure 3 shows the numbers of clusters in Dataset A obtained by DBSCAN and RADBSCAN. In RADBSCAN, the number of clusters is very stable, and the influence of *eps* is very small, which shows that our algorithm is not as *eps* sensitive as DBSCAN. In addition, RADBSCAN detect 7 clusters, and the real number of clusters is 5. The actual five clusters are detected and labeled manually. In fact, it can be concluded from table 9 that the 7 clusters found by the algorithm are more reasonable than the 5 clusters marked manually. The reasons will be described in detail in the following introduction to table 9.

Eps is a very important parameter in DBSCAN. When the value of *eps* increases from 0, the algorithm can find more clusters. This is because if *eps* is very small, only those clusters with high densities can be recognized. As shown in Figure 3, the number of clusters found by DBSCAN increases with increasing *eps*. DBSCAN finds more than 20 clusters, which is much more than the actual topic number. In our RADBSCAN algorithm, some clusters with the same topic are merged because of the operation of Algorithm 2 RegionQuery. The concept of our work is similar to GSDMM [32], which needs to define a large *K*, and we generate it automatically in our work according to density. Figures 4 and 5 show the number of clusters in Dataset B and Dataset C obtained by DBSCAN and RADBSCAN; RADBSCAN finds the correct number of clusters. According to Figure 4 and 5, the numbers of topics in dataset B and C detected by DBSCAN are much smaller than that in dataset A. Because, topic of smog is removed. Through the keyword extraction, we can find that the smog topic has many sub topics.

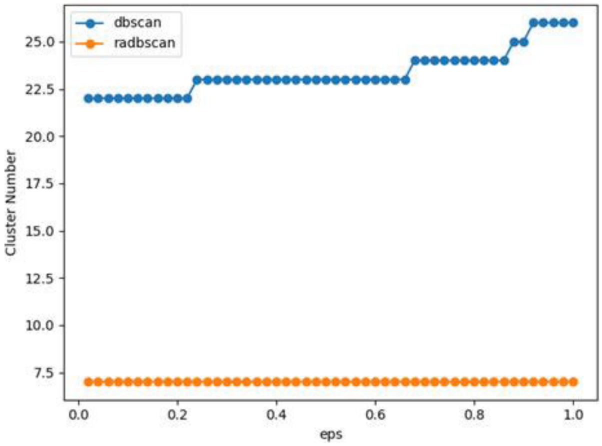


FIGURE 3. Numbers of clusters from DBSCAN and RADBSCAN based on Dataset A.

Figure 6 shows the NMI of two algorithms on Dataset A. Figure 6 shows that our clustering results are closer to the real clusters. The NMI value in Figure 6 appears to be a very small because of the noise recognition characteristics of DBSCAN and RADBSCAN. Some points far from the cluster centers are recognized as noise, but this does not affect the effect of the topic recognition. Our algorithm and DBSCAN both have disappointing NMI results. However, as shown in Figure 3 and Figure 6, we believe the advantage of our algorithm is that it avoids the influence of *eps*. In Figure 4,

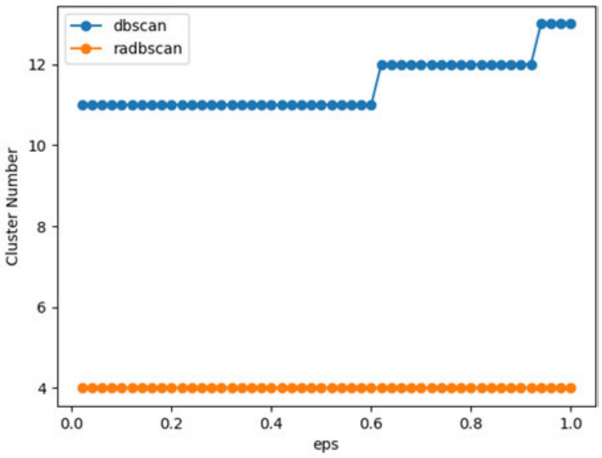


FIGURE 4. Numbers of clusters from DBSCAN and RADBSCAN based on Dataset B.

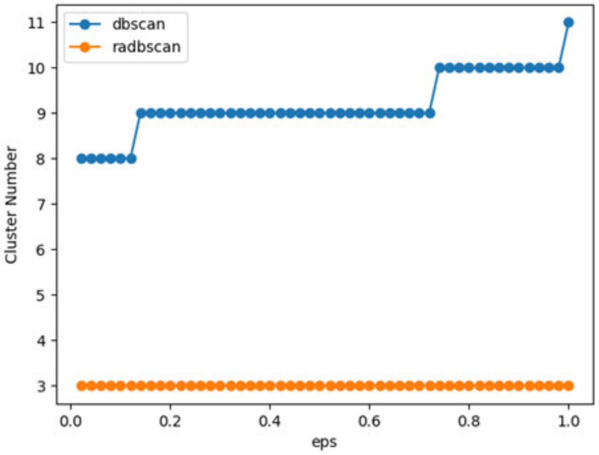


FIGURE 5. Numbers of clusters from DBSCAN and RADBSCAN based on Dataset C.

our NMI increases slightly because a small amount of noise enters the cluster, but this does not affect the final results of the topic detection. Figure 7 and Figure 8 show the NMIs of the two algorithms on Datasets B and C. Most of time, NMI increases a little with EPS, but sometimes it does not. Because larger EPS makes some noise classified correctly, but at the same time, it increases the number of clusters.

Table 9 shows the keywords extracted from these clusters. The topics of these clusters are very clear. The first, second, third and seventh clusters are about the Meizu cellphone, rocket team, civil servants and house prices, respectively. The fourth cluster has three keywords: air-blast, square and army green. This cluster describes a remarkable event, the appearance of army green air-blast machines in the square. Smog is a subject of public discontent in China, so the first appearance of an air-blast sprayer attracted great attention. It is believed that this invention can control smog. The news is still visible on Chinese websites. The sixth cluster is about fog and wind. It is hard for ordinary people to distinguish between natural fog and pollution smog, because they are often expressed with the same word ’雾’ in Chinese. Our algorithm separates these two topics. As shown from the clustering results, our algorithm not only finds all the topics but also has a better effect than manual labels.

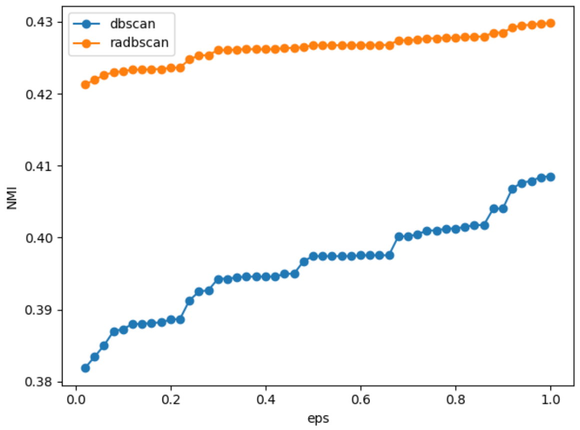


FIGURE 6. NMIs from DBSCAN and RADBSCAN based on Dataset A.

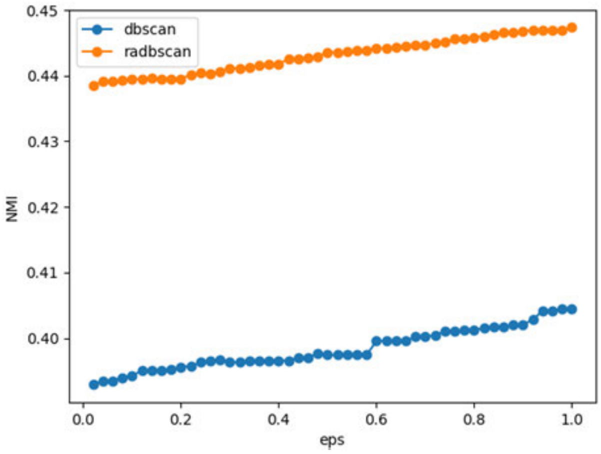


FIGURE 7. NMIs from DBSCAN and RADBSCAN based on Dataset B.

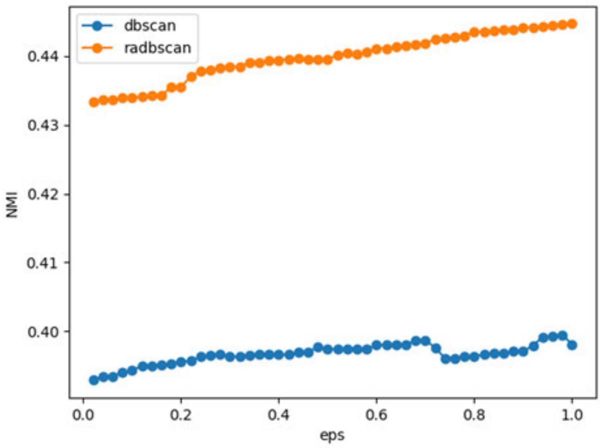


FIGURE 8. NMIs from DBSCAN and RADBSCAN based on Dataset C.

TABLE 9. Keywords extracted from clusters based on Dataset A.

Cluster	Keywords
1	Meizu
2	Rockets
3	Civil servants
4	Air-blast; Square; Army green
5	Smog
6	Wind; Fog
7	House price

TABLE 10. Clusters found by GSDMM and KSIHK on Dataset A.

	Initial number of clusters(K)	Number of clusters founded	Actual number of clusters	NMI
GSDMM	20	20	5	0.63
	10	10	5	0.65
	7	7	5	0.68
KSIHK	20	12	5	0.67
	10	6	5	0.69
	7	4	5	0.70

TABLE 11. Clusters found by GSDMM and KSIHK on Dataset B.

	Initial number of clusters(K)	Number of clusters founded	Actual number of clusters	NMI
GSDMM	20	18	4	0.62
	10	10	4	0.66
	7	7	4	0.71
KSIHK	20	11	4	0.66
	10	6	4	0.70
	7	3	4	0.71

Compared with other text clustering based topic detection works, K-free is one of our most successful endeavors. To prove that our algorithm is better than other algorithms, we choose two K-free algorithm, GSDMM [32] and KSIHK [33], to perform the same work on the all three datasets. GSDMM is a collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model for short text clustering. GSDMM needs to establish an initial K before executing; then, it finds the actual cluster number, which will be smaller than the initial K. KSIHK is a three-layer hybrid clustering algorithm designed for microblog topic detection. KSIHK also needs an initial K and finds clusters by k-mean; then, it merges small clusters of the same topic; finally, the number of clusters will converge to the true value. Table 10, Table 11 and Table 12 show the results. It is difficult for GSDMM and KSIHK to converge the number of clusters to the correct result.

TABLE 12. Clusters found by GSDMM and KSIHK on Dataset C.

	Initial number of clusters(K)	Number of clusters founded	Actual number of clusters	NMI
GSDMM	20	14	3	0.68
	10	9	3	0.63
	7	7	3	0.66
KSIHK	20	10	3	0.64
	10	4	3	0.62
	7	1	3	0.59

V. CONCLUSION

In our study: (1) we presented PANM, a neural attention model for sentence embedding. (2) We also proposed RADBSCAN, an improved density based approach for microblog sentence clustering.

In contrast to baseline methods, our approach has two advantages: (1) our sentence embedding method PANM performs better in sentence clustering task. Experiments on real-world datasets show that attention layer and p-mean layer could improve the capability of our neural network. (2) Our clustering algorithm can detect topics correctly and find the number of topics automatically from a microblog dataset. Experimental studies show that RADBSCAN can achieve significantly better performance than the baseline methods, which demonstrate it is effective to make use of forwarding relationship of microblogs.

Our work provides a simple method of detecting topics from microblogs without any prior knowledge, and it can help researchers make better use of microblog data. However, there is still a limitation to our approach. Our clustering algorithm makes use of microblog forwarding relationship, which requires users to be active and willing to forward microblogs. It makes our method more suitable for some popular social networking sites, such as Sina Weibo. We will solve it in the future.

Besides, we will continue to improve the performance of our work in the future. Two issues remain to be solved: (1) the presence of noise lowers the NMI indicator; (2) identifying the hierarchical structure of clusters.

REFERENCES

[1] T. Ma, Y. Zhao, H. Zhou, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "Natural disaster topic extraction in sina microblogging based on graph analysis," *Expert Syst. Appl.*, vol. 115, pp. 346–355, Jan. 2019.

[2] C. Wu, F. Wu, M. An, Y. Huang, and X. Xie, "Neural news recommendation with topic-aware news representation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1154–1159.

[3] X. Zhou, B. Wu, and Q. Jin, "Analysis of user network and correlation for community discovery based on topic-aware similarity and behavioral influence," *IEEE Trans. Human Mach. Syst.*, vol. 48, no. 6, pp. 559–571, Dec. 2017.

[4] A. Sasaki, K. Hanawa, N. Okazaki, and K. Inui, "Other topics you may also agree or disagree: Modeling inter-topic preferences using tweets and matrix factorization," in *Proc. Meeting Assoc. Comput. Linguistics*, Dec. 2017, pp. 398–408.

[5] H. Amoualian, W. Lu, E. Gaussier, G. Balikas, M. R. Amini, and M. Clausel, "Topical coherence in LDA-based models through induced segmentation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1799–1809.

[6] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An unsupervised neural attention model for aspect extraction," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 388–397.

[7] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 3857–3867.

[8] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis," *Cognit. Comput.*, vol. 10, no. 4, pp. 639–650, Aug. 2018.

[9] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning based text classification: A comprehensive review," 2020, *arXiv:2004.03705*. [Online]. Available: http://arxiv.org/abs/2004.03705

[10] M. Iyyer, A. Guha, S. Chaturvedi, J. Boyd-Graber, and H. Daumé, III, "Feuding families and former friends: Unsupervised learning for dynamic fictional relationships," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1534–1544.

[11] F. Huang, S. Zhang, J. Zhang, and G. Yu, "Multimodal learning for topic sentiment analysis in microblogging," *Neurocomputing*, vol. 253, pp. 144–153, Aug. 2017.

[12] S. Yang, G. Huang, and B. Cai, "Discovering topic representative terms for short text clustering," *IEEE Access*, vol. 7, pp. 92037–92047, 2019.

[13] C. Jia, M. B. Carson, X. Wang, and J. Yu, "Concept decompositions for short text clustering by identifying word communities," *Pattern Recognit.*, vol. 76, pp. 691–703, Apr. 2018.

[14] Y. Gu, Z. Qian, and F. Chen, "From Twitter to detector: Real-time traffic incident detection using social media data," *Transp. Res. C, Emerg. Technol.*, vol. 67, pp. 321–342, Jun. 2016.

[15] W. Magdy and T. Elsayed, "Unsupervised adaptive microblog filtering for broad dynamic topics," *Inf. Process. Manage.*, vol. 52, no. 4, pp. 513–528, Jul. 2016.

[16] Y. Chen, Y. Lv, X. Wang, L. Li, and F.-Y. Wang, "Detecting traffic information from social media texts with deep learning approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3049–3058, Aug. 2019.

[17] 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*, 2016, pp. 251–258.

[18] S. Xu, P. Li, F. Kong, Q. Zhu, and G. Zhou, "Topic tensor network for implicit discourse relation recognition in chinese," in *Proc. 57th Annu. Meeting Assoc. for Comput. Linguistics*, 2019, pp. 608–618.

[19] C. Li, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Topic modeling for short texts with auxiliary word embeddings," in *Proc. 39th Int. ACM SIGIR Conf.*, 2017, pp. 165–174.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, 2017, pp. 1–8.

[21] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 7370–7377.

[22] K. Skianis, F. Malliaros, and M. Vazirgiannis, "Fusing document, collection and label graph-based representations with word embeddings for text classification," in *Proc. 12th Workshop Graph-Based Methods Natural Lang. Process.*, 2018, pp. 1–5.

[23] W. Yang, J. Boyd-Graber, and P. Resnik, "A discriminative topic model using document network structure," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 686–696.

[24] C. Fu, Y. Du, B. Lyu, Q. Zhou, R. Hu, P. Jia, and Y. Zhou, "Forwarding behavior prediction based on microblog user features," *IEEE Access*, vol. 8, pp. 95170–95187, 2020.

[25] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowledg Discovery Data Mining*, 1996, pp. 226–231.

[26] J. Qiang, Z. Qian, Y. Li, Y. Yuan, and X. Wu, "Short text topic modeling techniques, applications, and performance: A survey," *IEEE Trans. Knowl. Data Eng.*, early access, May 4, 2020, doi: 10.1109/TKDE.2020.2992485.

[27] R. J. Gallagher, K. Reing, D. Kale, and G. Ver Steeg, "Anchored correlation explanation: Topic modeling with minimal domain knowledge," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 529–542, Dec. 2017.

[28] R. Hida, N. Takeishi, T. Yairi, and K. Hori, "Dynamic and static topic model for analyzing time-series document collections," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 516–520.

[29] F. Gurcan and N. E. Cagiltay, "Big data software engineering: Analysis of knowledge domains and skill sets using LDA-based topic modeling," *IEEE Access*, vol. 7, pp. 82541–82552, 2019.

[30] X. Ni, X. Quan, Z. Lu, L. Wenyin, and B. Hua, "Short text clustering by finding core terms," *Knowl. Inf. Syst.*, vol. 27, no. 3, pp. 345–365, Jun. 2011.

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

[32] J. Yin and J. Wang, "A Dirichlet multinomial mixture model-based approach for short text clustering," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 233–242.

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

[34] F. Yi, B. Jiang, and J. Wu, "Topic modeling for short texts via word embedding and document correlation," *IEEE Access*, vol. 8, pp. 30692–30705, 2020.

[35] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, and I. Gurevych, "Classification and clustering of arguments with contextualized word embeddings," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1–5.

[36] S. A. Curiskis, B. Drake, T. R. Osborn, and P. J. Kennedy, "An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit," *Inf. Process. Manage.*, vol. 57, no. 2, Mar. 2020, Art. no. 102034.

[37] W. Zhang, Y. Li, and S. Wang, "Learning document representation via topic-enhanced LSTM model," *Knowl.-Based Syst.*, vol. 174, pp. 194–204, Jun. 2019.

[38] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," *Comput. Sci.*, vol. 1, no. 4, pp. 39–44, 2015.

[39] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1422–1432.

[40] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–8.

[41] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 670–680.

[42] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018, pp. 1–5.

[43] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, Valencia, Spain, 2017, pp. 427–431.

[44] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, 2015, pp. 1681–1691.

[45] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: http://arxiv.org/abs/1408.5882

[46] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," 2015, *arXiv:1510.03820*. [Online]. Available: http://arxiv.org/abs/1510.03820

[47] H. Zhang, L. Xiao, Y. Wang, and Y. Jin, "A generalized recurrent neural architecture for text classification with multi-task learning," in *Proc. Twenty-Sixth Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2873–2879.

[48] M. Yang, W. Zhao, J. Ye, Z. Lei, Z. Zhao, and S. Zhang, "Investigating capsule networks with dynamic routing for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3110–3119.

[49] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[50] Z. Lin, M. Feng, C. N. Dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, 2017, pp. 1–5.

[51] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Barcelona, Spain, Jul. 2004, pp. 404–411.

[52] X. Zhou, W. Liang, K. I.-K. Wang, R. Huang, and Q. Jin, "Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data," *IEEE Trans. Emerg. Topics Comput.*, early access, Jul. 26, 2019, doi: 10.1109/TETC.2018.2860051.

[53] X. Zhou, W. Liang, K. I.-K. Wang, and S. Shimizu, "Multi-modality behavioral influence analysis for personalized recommendations in health social media environment," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 5, pp. 888–897, Oct. 2019.

[54] A. Rácklé, S. Eger, M. Peyrard, and I. Gurevych, "Concatenated power mean word embeddings as universal cross-lingual sentence representations," 2018, *arXiv:1803.01400*. [Online]. Available: http://arxiv.org/abs/1803.01400

[55] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Trans. Assoc. Comput. Linguistics*, vol. 2, pp. 207–218, Dec. 2014.

[56] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," 2015, *arXiv:1511.08198*. [Online]. Available: http://arxiv.org/abs/1511.08198

CONG WAN received the B.Eng. degree from Northeastern University at Qinhuangdao, Qinhuangdao, China, the M.S. degree from Northwestern Polytechnical University, China, and the Ph.D. degree from Northeastern University. He is currently a Lecturer with the School of Computer and Communication Engineering, Northeastern University at Qinhuangdao. His main research interests include complex networks, neural networks, resource allocation in cloud computing, and so on.

SHAN JIANG was born in Dalian, China. She received the B.Eng. degree from the Tianjin University of Technology, China. She is currently pursuing the master's degree with Northeastern University, China.

CONG WANG was born in Qinhuangdao, China, in 1981. He received the B.Eng. and M.S. degrees in computer science from Yanshan University, Qinhuangdao, in 2004 and 2008, respectively, and the Ph.D. degree in computer application technology from Northeastern University at Shenyang, Shenyang, China.

He is currently a Lecturer with the School of Computer and Communication Engineering, Northeastern University at Qinhuangdao, Qinhuangdao. His main research interests include virtual network embedding, resource allocation in cloud computing, and so on.

YING YUAN received the B.Eng. degree from Northeastern University, the M.S. degree from Yanshan University, China, and the Ph.D. degree from Northeastern University. She is currently a Lecturer with the Computer Center, Northeastern University at Qinhuangdao. Her main research interests include virtual network embedding and virtual resource allocation in cloud computing.

CUIRONG WANG received the Ph.D. degree from Northeastern University, China, in 2006. She is currently a Full Professor with the School of Computer Science and Engineering, Northeastern University. Her research interests include cloud computing, big data analysis, and distributed computing. She is a Senior Member of CCF.