IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 24, 2020, accepted December 7, 2020, date of publication December 25, 2020, date of current version January 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047458

Topic Detection and Tracking Based on Windowed DBSCAN and Parallel KNN

CHUANZHEN LI[1], MINQIAO LIU[1], JUANJUAN CAI[2], YANG YU[3], AND HUI WANG[4]

[1]School of Information and Communication Engineering, Communication University of China, Beijing 100024, China
[2]Key Laboratory of Media Audio and Video (Communication University of China), Ministry of Education, Communication University of China, Beijing 100024, China
[3]iQIYI Inc., Beijing 100080, China
[4]State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China

Corresponding author: Hui Wang (hwang@cuc.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2020YFF0305300.

ABSTRACT Topic Detection and Tracking technique (TDT) has been commonly used to identify the hot topics from the huge volume of Internet news information and keep up with the hot news. However, traditional topic detection and tracking methods have shown low accuracy and low efficiency. In this paper, a topic detection system driven by big data is built on the Spark platform, which aims at improving the efficiency of news collecting from the Internet and improving the accuracy and efficiency of topic detection and tracking tasks. This system can be easily employed in a distributed architecture and work as a parallelized news collecting and topic detection system. An improved density-based spatial clustering of application with noise (DBSCAN) clustering algorithm based on the time window is proposed to achieve accurate topic detection with the auxiliary advantage of reducing the time complexity. A parallel KNN based topic tracking algorithm is proposed for the topic tracking task. Experiments including comparison with some baseline algorithms and quantitative and qualitative analyses are conducted on pseudo-distributed Spark platform, which demonstrates the effectiveness and efficiency of the parallelized topic detection system.

INDEX TERMS Big data, DBSCAN, parallelized, TDT.

I. INTRODUCTION

With the rapid paradigm shift of information access, news and information could be provided by online news websites, mainstream media, and individual users as well. With the improvement of network coverage, more and more Internet users publish, forward, and comment on the latest breaking news on social media sites. However, it is difficult to discover hot topics quickly and track them in time because of the explosive growth in the amount of Internet data. Topic Detection and Tracking (TDT) technique has been commonly adopted to help people acquire the instant news and keep their eyes on the hot news.

Generally, a TDT system normally consists of data collection layer and data analysis layer, as shown in Figure 1. Data collection layer is mainly responsible for the extraction and preprocessing of the news data, which serves as the data source of the hot topic detection system. The preprocessing module is responsible for the filtering of the news which has missing components or properties.

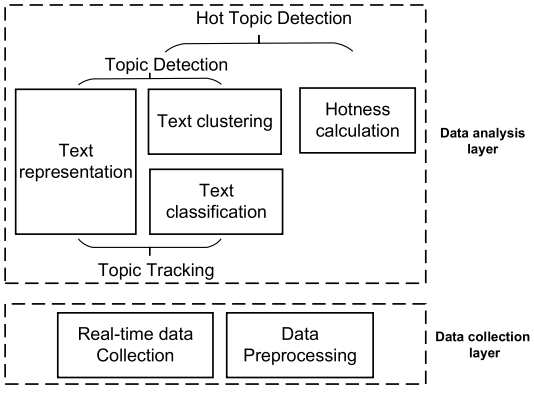

FIGURE 1. Framework of the hot topic detection system.

Data analysis layer is responsible for topic detection, topic hotness calculation, and topic tracking. In the data analysis module, topic detection is usually conducted through text clustering on massive news data to recognize news topics.

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

Then, the news hotness index or news popularity index is calculated to assign a score to each topic cluster. Topic tracking is mostly applied to identify the news related to a certain topic. If users are interested in the hotness development trend, the hotness trend map of the topics is also provided by the topic tracking task. Topic tracking is usually considered as classification problem to assign a topic to the nearest cluster center. It is worth mentioning that text representation is the foundation of topic detection task, and topic tracking task as well.

With the blooming of news websites and social media, traditional approaches are not satisfying due to the explosive growth of data. There are two cons of these traditional methods: low accuracy and low efficiency.

A good representation is helpful to improve accuracy of topic detection and tracking, and word embedding based methods [1], [2] which automatically learn low-dimension dense vectors for texts outperform traditional methods such as Vector Space Model (VSM) and Relevance Model [3]–[5]. Some literature improves the accuracy of TDT tasks by means of the extern knowledge databases such as the Knowledge Graph, synonym dictionary [6], [7]. Besides, variety of improved clustering algorithms have been designed aiming at getting more accurate and explainable clustering results [8]–[15].

Improving the efficiency of models or algorithms is as important as raising accuracy. Topic detection and tracking heavily rely on the clustering algorithms with different complexities. To improve the efficiency of these clustering algorithms, current researches mainly focus on parallelizing the TDT algorithms by big data tools [16], [17]. In addition, a fine-tune on initialization settings of clustering algorithms and well-designed rules for clustering procedure [18]–[21] can advance the execution efficiency of clustering algorithms.

In this research, to ensure a high level of accuracy in TDT tasks, the double vectors text representation model puts much emphasis on meaningful named entity and terms in the titles to extract more exact text representations, since that not all words in news texts have equal importance and the title of the news is a refinement of the body text of the news which is much related to the topic. Moreover, facing the challenge of high time complexity in the process of text clustering, the proposed model is based on the time windowed density-based spatial clustering of application with noise (DBSCAN) algorithm and big data platform. In such settings, the time complexity can be reduced from $O(n^2)$ to $O(n)$ as analyzed in latter section.

The main work of this paper is as below:

1) An improved DBSCAN clustering algorithm based on the time window is proposed, and it adopts an implementation of parallelization to process a huge amount of data stream.
2) The double vectors text representation model which consists of the title vector and content vector of the news is used to represent the news texts for accurate topic detection tasks. Particularly, by introducing two hyper-parameters the proposed double vectors text representation model can capture more distinct information about the news itself.
3) The optimized parallelized KNN method is applied to the topic tracking task by utilizing a two-stage clustering procedure. It ensures a high score of precision on the topic tracking task. Moreover, the timestamp feature is taken into consideration to reduce the computation cost.

II. RELATED WORK

To conduct the task of topic detection and tracking effectively, text representation and various text classification algorithms have been applied to the framework.

A. TEXT REPRESENTATION

Before the process of the text clustering and text classification, it is of necessity to vectorize the texts. Text representation algorithms can be commonly classified into the below categories.

1) VECTOR SPACE MODEL

Vector Space Model (VSM) [3] is one of the dominant models used to learn text representations. VSM is firstly proposed by Salton *et al.* [3] to be used in the field of information retrieval. Specifically, in document retrieval area, VSM is proposed to represent a document by a multi-dimensional vector, and each element represents a different term in the document. Given a common document, it can be represented as below:

$$dj = (w_{1,j}, w_{2,j}, \ldots, w_{i,j}, \ldots, w_{n,j}), \tag{1}$$

where $w_{i,j}$ represents the *i-th* feature of the document j, and n is the number of the features belonging to the document j.

The widely used method of computing the weight of features is Term Frequency-Inverse Document Frequency (TF-IDF) [22]. TF-IDF is a feasible method to compute the importance of words in the news texts, and it utilizes the product of the Term Frequency (TF) and Inverse Document Frequency (IDF) to represent the feature word t in document d, which is defined as below:

$$\text{TF-IDF}(d, t) = \text{TF}(d, t) \times \text{IDF}(t), \tag{2}$$

where TF(d, t) is the term frequency of word t in the specified document d, and the IDF(t) indicates that how discriminative a term is in distinguishing the current document from other documents. The VSM model for text similarity is pretty straight-forward. However, relationships between the features are discarded when the text is transferred into a feature vector due to the independence assumption that text features are independent of each other.

Language Model and Relevance Model are two types of probabilistic models [4], [5], [23] to consider the semantic similarities between the words. To take the relationships between words into account and achieve semantics fusion, some literature makes use of the extern knowledge databases such as the Knowledge Graph, synonym dictionary [6], [7].

However, the dimension of the text vectors grows larger as the size of the corpus becomes bigger, thus resulting in poor efficiency of computation.

2) WORD EMBEDDING MODEL

Word embedding is a low-dimensional real value vector based on the distributional hypothesis, namely, words with the same semantics are closer to each other in the embedding space. The constraint of independence among all the features and the curse of dimensionality in the VSM can be eliminated by using low-dimensional word embedding [1], [2].

Word2vec is one of the word embedding models [1], [24]. Word embedding is a shallow neural network model, and can be trained like the neural network. The words which have similar meaning may get a higher cosine similarity in the word embedding model. In the word embedding model, each word is a point in the embedding space which also represents the semantics information. Thus the semantics similarity can be computed by the cosine similarity metrics.

For a certain text which is made up of a set of words $W = \{w_1, w_2, \ldots, w_i, \ldots, w_N\}$, the word embedding model represents it as the form below:

$$text\_vec = \frac{1}{N} \times \sum_{i=0}^{N} d_{w_i} \times weight_i, \tag{3}$$

where the *text_vec* stands for the vectorization representation of a text, d_{wi} and $weight_i$ are the word embedding and weight of the word w_i respectively, and N is the number of the different words contained in the text W.

B. TEXT CLUSTERING ALGORITHM

The goal of the text clustering is to group the text into several clusters according to some properties or features they share. Specifically, the goal of the text clustering in the topic detection tasks is to merge the texts related to the same topic into a cluster.

In general, the clustering algorithms can be divided into two categories, namely, clustering algorithms based on the number of clusters such as the K-Means algorithm of which the time complexity is O(n), and the clustering algorithms based on the threshold of similarity metrics such as Single-Pass clustering and DBSCAN of which the time complexity is O(n^2).

Among the clustering algorithms, the relatively simple one is the hierarchical clustering algorithm [25] which has a fatal weakness that the computation efficiency decreases with the increasing text vector length.

To improve the quality of the clustering results, a lot of improved K-Means clustering algorithms focus on the similarity computation [18], [26], [20]. Liu *et al* [10] proposed a feature selection method called ''K-Means based feature selection'' which selects informative features by applying different feature selecting methods to diverse clustering results. Geoff Hulten *et al.* optimized the clusters by applying some clustering strategies and can process all available data in parallel [27].

The common clustering algorithms used in the topic detection are Single-Pass and DBSCAN. To avoid the influence of the sequence of text input and high time complexity as shown in the K-Means clustering, Yan *et al.* [11] improve the Single-Pass algorithm by applying a two-step Single-Pass clustering process to the micro-blog dataset with a time window. Zhou *et al.* [12] propose an improved Single-Pass clustering algorithm using double similarity thresholds which represent the similarity between sub-topics under a topic and similarity between different topics respectively to get a lower missed detection rate and false detection rate. In this paper, the DBSCAN is adopted as the basic clustering algorithm for the topic detection task. In general, DBSCAN is a density-based clustering algorithm of which the main definition related is defined as below:

(1) ε-neighborhood of a point: The ε-neighborhood of a point p is referred to as a set of points EN(p) in which the distance between an element of EN(p) and the target point p is no more than ε which can be defined by manual.

(2) core point: The point of which the ε-neighborhood has a size bigger than *MiniP* is a core point, where the *MiniP* is a value that belongs to an integer.

(3) Density-reachable: if a point p is a core point meanwhile another point q has a distance less than ε with a point in the ε-neighborhood of point p, then the point q is density-reachable from p.

(4) Density-connected: if both p and q are density-reachable from point w, then p is density-connected to q or q is density-connected to p.

The main steps of DBSCAN algorithm can be presented in Algorithm 1.

Note that the function generator() is used to randomly generate a cluster id at each iteration, and the cluster id is set to NOISE before the first loop of iterations and the cluster id of a vector can be changed later if the vector is density-reachable from a core point.

To correct the clustering result of Single-Pass for hot topic detection, the clustering result is updated by DBSCAN algorithm regularly in [28]. According to the fact that newly developing topics is far away from the high-density region where normal topic clusters locate, the momentum based topic detection model with DBSCAN algorithm [29] is proposed to find out newly developing topics by detecting outliers in the microblog data. Yang [30] proposes an incremental DBSCAN clustering algorithm for topic detection to deal with incremental microblog data, this algorithm can satisfy the demand of real-time topic detection. Given the difficulties in parameter determination, Sun and Liu [31] improve DBSCAN algorithm with a dynamic parameter adjustment scheme, and achieves good performance in hot topic detection by decreased missed detection rate and false detection rate, but the time consuming in a single iteration is not that ideal.

C. TEXT CLASSIFICATION ALGORITHM

Topic tracking can be regarded as a classification problem. KNN and its improved algorithms are widely used in topic

Algorithm 1 DBSCAN Algorithm

INPUT: the feature vectors of the texts $SetPts = \{S0, S1, S2, \ldots, Si, \ldots\}$, ε, $MinPoints$

OUTPUT: A set of clusters $ExiClu = \{Cl0, Cl1, Cl2, \ldots, Cli, \ldots\}$

1: Set the cluster id of all the feature vectors in *SetPts* to *UnKnown*.
2: Create a set of clusters named *ExiClu* which is an empty set initially.
3: *clusterid*: = generator(*NOISE*);
4: **for** *i* in *SetPts*:
5: **if** *i*. *ClusterId* == *UnKnown*
6: **if** GenCluster(*SetPts*, *i*, ε, *clusterid*, *MinPoints*)
7: *clusterid*: = generator(*ID*);
8: Merge the feature vectors which share the same *clusterid* into the same cluster *Cli*, and then add these clusters to the set *ExiClu*.
9: **return** *ExiClu*;
10: **Function** GenCluster(*SetPts*, *Pt*, *CId*, ε, *MPs*):
11: *NeigofPt*: = GetNeigofPt(*Pt*, ε);
12: **if** sizeof(*NeigofPt*) < *MPs*
13: Set the cluster id of the *Pt* to *NOISE*;
14: **return** *false*;
15: **else**
16: Change the Cluster id of all points in the *NeigofPt* to *CId*;
17: Delete the point *Pt* from the *NeigofPt*.
18: **while** (sizeof(*NeigofPt*) > 0):
19: Randomly select an element *PointNei* of *NeigofPt*.
20: *NeigofPNei* = GetNeigofPt(*PointNei*, ε);
21: **if** sizeof(*NeigofPNei*) >= *MPs*
22: **for** *j* in *NeigofPNei*:
23: **if** *j*.*ClusterId* == *UnKnown*
24: Add the point *j* to the *NeigofPt*;
25: Set the cluster id of *NeigofPNei* to *CId*;
26: Delete the point *PointNei*;
27: **return** *true*;

tracking tasks [32]. KNN methods compute the correlation of a new text with training dataset at first, then select constant number K training samples which are related to the text most, and classify the text into a topic cluster according to the K nearest samples. Carbonell *et al.* [14] adopt decision tree and KNN to track topics. Diao *et al.* [33] consider the finite life period of a news text and only accept the news texts within the time window of a topic, then classify the news texts by comparing the difference of averaged similarity with *k1* positive samples and that with *k2* negative samples. Zhang *et al.* [34] compute the averaged value of similarity with positive or negative samples to smooth the classification results, and add two adaptive threshold parameters to avoid the issue of theme drift and improve the accuracy of topic tracking. To decrease the computation complexity, Chen [15] applies improved KNN to the topic tracking tasks of the multi-resource data by computing the similarity between the centroids of different kinds of topics and the news report to be tracked. Relative Weight KNN [35] is proposed to solve the problem of data bias in the training dataset, and it considers the rank of each K-nearest neighbor and the category distribution of training data simultaneously to assign every testing sample a more accurate category label.

D. ALGORITHM PARALLELIZATION

With the continuous growth of news data, traditional serial TDT algorithms suffer from the problem of high time complexity. As a result, algorithm parallelization for TDT has become one of the research priorities. Generally, the algorithms are parallelized through the frameworks for processing huge datasets, such as Hadoop and Spark.

Wang [16] investigates the hot topic extraction system which is built on Spark and improves the K-Means algorithm with a parallelization implementation. Aiming at solving the problem of having difficulty in getting valuable hot topics from a huge amount of digitized textual materials quickly and accurately, Ai and Li [17] propose a parallel two-phase hot topic detection algorithm. The fine-grained similarity computation method is exploited to remove the noisy items at the first phase. And the single-pass algorithm with coarse-grained similarity computation is developed to produce the final hot topics set at the second phase. All these two phases are implemented with Spark on the cloud.

In this paper, by using the Spark platform, the proposed Topic Detection System driven by Big Data (TDSBD) can be employed on multiple computers to work efficiently with the advantage of high throughput of the daily news texts on the Internet.

III. TIME WINDOW BASED PARALLELIZED TOPIC DETECTION

The paper proposes a parallelized topic detection approach. Specifically, the text representation for the topic detection is designed specially in a parallel way, and is improved using the common features of the texts; the time window based parallelized DBSCAN clustering algorithm is proposed and is analyzed from the perspective of time complexity. Then the implementation of the parallel topic detection via Spark is made. Finally, the hotness computation formula is discussed in brief.

A. THE OVERALL PROCESS OF PARALLELIZED TOPIC DETECTION

The basic flow chart of the time window based parallelized topic detection is shown as in Figure 2. It includes text representation, vector merging, time window based clustering and clusters merging.

The process of the time window based parallelized topic detection mainly consists of the following steps:

1) Distribute the news texts to each slave server of the Spark cluster which consist of several servers connected to the Master server, then the word embedding

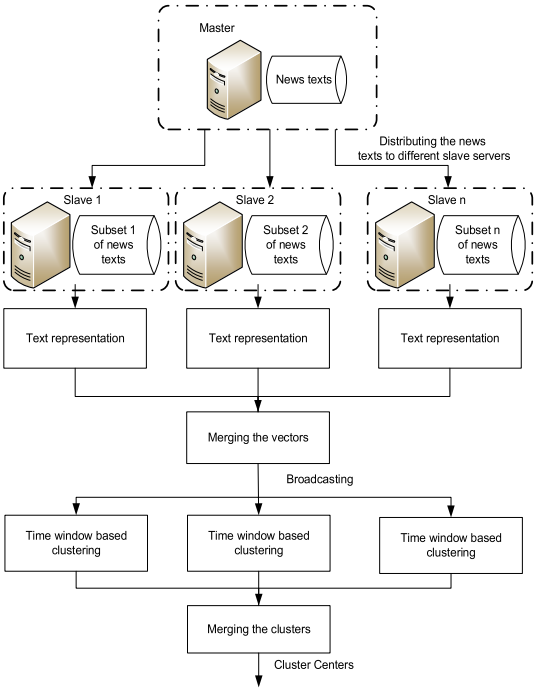

FIGURE 2. Time window based parallelized topic detection.

model pre-trained on a huge corpus and the computed IDF value of each word are broadcast to all the slave servers.

2) All the slave servers simultaneously process the news texts by the operations of word split, stop words removal and counting the term frequency (TF) of each word in the news text, etc.; then the weights of the words are calculated. By the end of this step, the published time of each news is converted into the format of the timestamp.
3) All the slave servers represent the news texts as the double vectors text representations.
4) Merge the text representations and the published time of the news texts, and again broadcast them to all the slave servers. The slave server will compute similarities between the target news text and the news texts within its corresponding time window and exploit the time window based DBSCAN clustering algorithm to generate clusters of topics.
5) Merge all the clusters which are distributed in all the slave servers to form the final clusters of topics.

B. DOUBLE VECTORS TEXT REPRESENTATION MODEL

The news texts need to be separated to obtain meaningful words and expressions. In this paper, the preprocess of words split is conducted by the Chinese word split tool named Jieba with an extra user-defined dictionary incorporated to improve the accuracy of word split. The parts-of-speech of the words and expressions are labeled automatically with Jieba. The nouns, verbs, acronyms in Chinese, and English words and unregistered words are reserved for topic detection. Then, TF-IDF technique is adopted to compute the weights of the reserved words.

Some factors must be considered. Firstly, the contribution that each word makes to the classification of the topics is not the same. Intuitively, the titles, nouns, verbs and the named entities are more important and discriminative to the topic category classification, which is not considered in the weights of the TF-IDF technique. In view of this problem, in the paper, two hyper-parameters are added to the computation formula which is defined as below:

$$weight_{i,j} = \text{TF-IDF}(i,j) \times \alpha^t \times \beta^e, \tag{4}$$

where α and β are two hyper-parameters, and the former one is responsible for putting emphasis on the term i which appears in the title of the news text j with the parameter t being set to 1 otherwise 0, and the latter one is responsible for emphasizing the term i which is a named entity in the news text j with the parameter e being set to 1 otherwise 0.

Secondly, different words may refer to the same named entity. The clustering algorithm may suffer from degraded quality due to the misjudgment about words that share the same named entity. To solve this problem, the word embedding is used to represent the news texts with the advantage of better semantics representation by the closer distance between the two terms which share the same named entity.

Formally, the double vectors text representation includes title vector and content vector of the news text. Since the title of the news is a refinement of the body text of the news, it is much related to the topic. The title vector is represented using the sentence embedding method which is a simple but more efficient method in the textual similarity tasks [36]. For simplicity, this method just computes the weighted average of the word vectors in a sentence and then remove the projections of the average vectors on their first singular vector. The weight of each word w in a sentence is weighted as $\frac{\alpha}{\alpha+p(w)}$, where the α is a hyper-parameter, $p(w)$ is the term frequency of w estimated from any corpus.

The content vector of the news j is a weighted sum of all the word embeddings of the body text of the news, which is defined as below:

$$content_j = \frac{1}{N} \times \sum_{i=1}^{N} word_{w_i} \times weight_{w_i,j}, \tag{5}$$

where the $word_{wi}$ and $weight_{wi,j}$ are the word embedding and weight of the i-th word in the body text of the news j respectively.

Similarly, the title vector and the content vector can be computed in a parallel way by distributing all the news text to the slave servers. Each slave server carries on word splitting, parts-of-speech tagging and stop words removal and so on. Then the title vector and content vector can be computed using the method in reference [36] and the equation (5) after computing the term frequency of the words in each news text. Meanwhile, the published time of each news is converted into the format of timestamp by the slave servers. The final text representation is made up of three parts, namely, the content

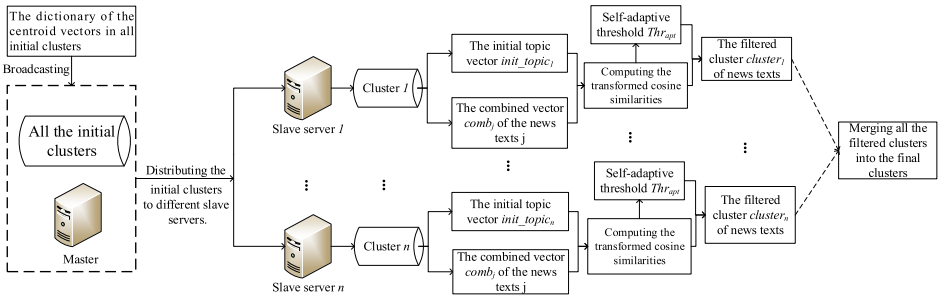

FIGURE 3. The process of filtering the initial clusters using a self-adaptive similarity threshold.

vector and the title vector, a timestamp calculated by the published time of the news. Then the master server merges all these three parts into three matrices, i.e., title vector matrix, content vector matrix and timestamp matrix. These matrices containing the global information of all the news texts will be broadcast to the slave servers for the task of clustering.

C. TIME WINDOW BASED SIMILARITY COMPUTATION

To overcome the problem of high time complexity in traditional clustering algorithms, an improved clustering method is achieved by computing the similarity between the target text located in the center of a finite size of time window and the news texts of which the timestamp is within the time window. Specifically, a specified news text is used to compute the similarity only with the texts of which the timestamp is within the interval of $[data\_time_i - T, data\_time_i + T]$, where T is an integer number to limit the length of the time window.

Since the number of the news texts that the new websites release every day is approximately constant, let the number of the news texts generated by the news websites every day be K. Assuming that the number of the news texts needed to be clustered into n different topics, the total number of the similarity computation will be $3(2T + 1)nK$ which results in a time complexity of $O(n)$. The speed of the algorithm execution is accelerated significantly because the workload of similarity computation is distributed to all slave servers to be executed in parallel.

Formally, for similarity computation, let one of the news texts be $\{key = doc\_id_i, value = (title\_vec_i, content\_vec_i, data\_time_i)\}$, each slave server passes through the news texts, and filters out the news texts of which the timestamp is within the interval of $[data\_time_i - T, data\_time_i + T]$, and extracts the corresponding title vectors, content vectors to form partial title vector matrix PT and content vector matrix PC. Then the similarity between the target news text and the news texts within the time window can be computed using the equation (10) which is defined as below:

$$sim(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}, \tag{6}$$

$$simt(x, y) = \begin{cases} (sim(x, y) - T_s) \times \gamma, & if\ sim(x, y) > T_s \\ 0, & if\ sim(x, y) \le T_s \end{cases} \tag{7}$$

$$comb_i = \alpha \times t_i + \beta \times c_i \tag{8}$$

$$\alpha + \beta = 1 \tag{9}$$

$$d\_simi(i, j) = \max\{simt(t_i, t_j), simt(c_i, c_j), simt(comb_i, comb_j)\} \tag{10}$$

Note that the *sim(x,y)* is the so-called cosine similarity metric, and *simt(x,y)* is a transformed cosine similarity where T_s is the similarity threshold, and γ is a scale factor. In equation (8) (9) (10), the c_i is the content vector of news text i, the t_i refers to the title vector of the news text i, $comb_i$ represents a combination of the title vector t_i and the content vector c_i. *simt(x,y)* introduces the nonlinearity by setting a similarity threshold T_s, thus making it more discriminative in the topic detection. Equation (10) indicates that the final similarity *d_simi(i,j)* is determined by the maximum of the title vector similarity, the content vector similarity and the combined vector similarity. Then each slave server takes out the news texts of which the similarity with the target news text is above the similarity threshold T_s as an initial topic cluster noted as $init\_cluster_i$. Then the corresponding initial topic vector $init\_topic\_vec_i$ is defined as below:

$$init\_topic\_vec_i = \frac{1}{n} \sum_{j \in init\_cluster_i}^{n=length(init\_cluster_i)} comb_j, \tag{11}$$

Finally, the text representation in every slave server has the form of $\{key = doc\_id_j, value = (init\_cluster_i, comb_j)\}$, then the initial topic vectors $init\_topic\_vec_i$ stored in all the slave servers are merged into a dictionary *cluster_dict* of the centroid vectors in all initial clusters.

In general, the topic density among the news texts is not well-distributed, which results in low accuracy of the clustering algorithm with a fixed similarity threshold. A better solution to mitigate this problem is to use a self-adaptive similarity threshold to filter out the related news texts from the initial clusters. Figure 3. shows the process of filtering the initial clusters with a self-adaptive similarity threshold. To start with, each slave server computes the transformed cosine similarities $simt(init\_topic\_vec_i, comb_j)$ between the initial topic vector $init\_topic\_vec_i$ and the news texts combined vector $comb_j$ using the equation (7), then the self-adaptive threshold

can be described as below:

$$simt\_aver_i = \frac{1}{n} \sum_{j \in init\_cluster_i} simt(init\_topic\_vec_i, comb_j), \tag{12}$$

$$Thr_{apt} = Func(T_c, simt\_aver_i), \tag{13}$$

where n is the number of the news texts in the topic cluster $init\_cluster_i$, $simt\_aver_i$ is the average of the transformed cosine similarities $simt(init\_topic_i, comb_j)$ with respect to the topic cluster $init\_cluster_i$. The Thr_{apt} is the self-adaptive threshold which is a function of T_c and $simit\_aver_i$, and T_c is a threshold parameter determined by the categories of the news texts. The $Func()$ is a function of which the form is needed to be explored. The parameter $simit\_aver_i$ can represent the density of certain topic i which has a positive relationship with the parameter $simit\_aver_i$. Hence, when the $simit\_aver_i$ is smaller, the density of the topic i is lower, the self-adaptive threshold should be getting bigger to exclude irrelevant samples. From this point, the simplest assumption with the $Func()$ is that the self-adaptive threshold has a linear negative correlation with the $simit\_aver_i$, which can be described as below:

$$Thr_{apt} = T_c - simt\_aver, \tag{14}$$

Since the form of the $Func()$ is determined, the self-adaptive threshold Thr_{apt} thus can be exploited to filter the news texts in the initial clusters to generate the final clusters.

In summary, the time window based DBSCAN clustering algorithm using double vectors text representation can be described as below:

1) The news texts are split into several small datasets in the master server, then these small datasets are distributed to all the slave servers.
2) Compute the double vectors and timestamp of each news text in the double vectors text representation model on the slave servers in parallel.
3) The time window based similarity computation is executed in all slave servers in parallel, and only the news text of which the timestamp is within the time window of the target news text can be involved into the similarity computation with the target text. Then the DBSCAN clustering is applied to the news texts to get the initial clusters.
4) A self-adaptive similarity threshold is computed to filter out the news texts which have more reliable and higher similarity and thus are more likely to share the same topic as the shaped topic clusters.

5) Merge the shaped topic clusters distributed in all the slave servers to obtain the collection of the ultimate topic clusters.

Note that the time window based DBSCAN algorithm can achieve high accuracy with high computation efficiency and low runtime complexity since the topic detection task is employed on the distributed framework.

Once the topic detection task is finished, the hotness computation formula can be applied to give a score for each topic cluster. The hotness computation formula used in this paper is the original formula proposed in reference [37] which is defined as below:

$$Hotness(j_t) = \sum_{s=1}^{K} |D_{js(t)}| \times \exp(\frac{D_{js(t)}}{N_{s(t)}} \times W_s) \times WB, \tag{15}$$

$$|Djs_{(t)}| = \frac{D_{js(t)}}{\sqrt{\sum_{c=1}^{C} D_{cs(t)}}} \tag{16}$$

where $Hotness(j_t)$ is the hotness value of the topic j within the time window t, and $D_{js(t)}$ is the number of the news texts which belong to the topic j from the news source s within the time window t, and K is the number of the news sources, C is the set of the topics. $|D_{js(t)}|$ is a normalized topic frequency of the topic j from the news source s within the time window t. $N_{s(t)}$ is the total number of the news texts from the news source s within the time window t. As for the W_s, it is a weight of the news source s, which will be 1 if the news source s is a news website, or will be 0.1 if the news source s is a BBS message source. WB is set to 2 if a topic appears both in news website and BBS message source, otherwise it is set to 1.

Since the news texts are all from the news websites in this paper, the parameter W_s in formula (15) is revised by the means of the ranking of the web ports including the BBS message boards and news websites ranked by the professional ranking website [38].

IV. THE PARALLEL KNN BASED TOPIC TRACKING ALGORITHM

Traditional topic tracking infers the topic label of the news texts mostly by its similarity with the training datasets. However, it has two disadvantages when processing a huge volume of news data. Firstly, it has poor scalability. When the training dataset gets larger, the similarity computation costs much time, thus resulting in low efficiency in topic tracking. Secondly, traditional KNN based topic tracking cannot respond to the streaming data of the news texts in time due to its single-machine implementation.

Given these problems in the traditional KNN based topic tracking, some optimizations are proposed. In this paper, only the similarities between the news text to be tracked and the samples of which the timestamp is within a limited time window are computed to reduce the computations. Furthermore, a two-stage topic tracking process is made. In the first stage, the time window based DBSCAN is applied to do the fine-grained topic detection with a higher similarity threshold. By doing so, the intra similarities between the elements of a sub-topic set can be very high. In the second stage, the news texts of a sub-topic set are filtered using another similarity threshold according to their similarity with the average vector of all the elements of the sub-topic set.

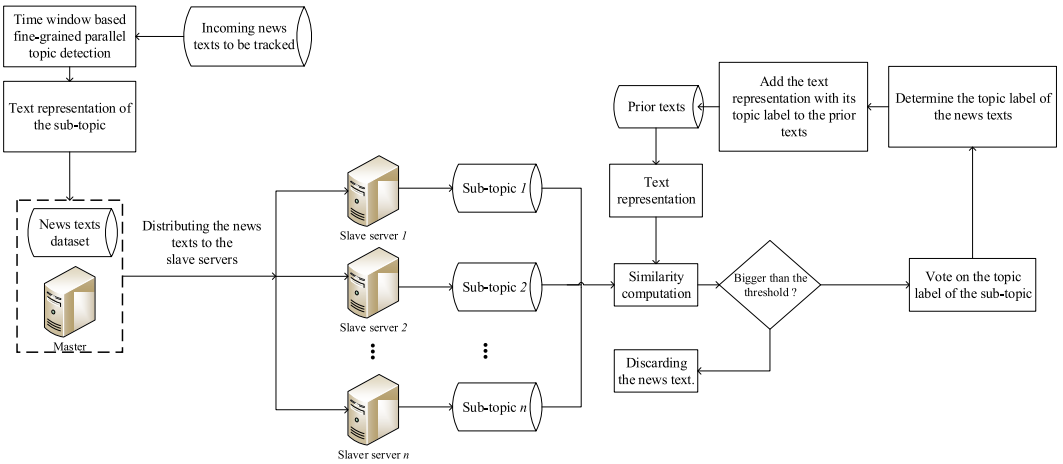

FIGURE 4. The processing of the proposed topic tracking algorithm.

A. THE PROCESSING OF THE PROPOSED TOPIC TRACKING ALGORITHM

As Figure 4. shows, the processing of the parallel KNN based topic tracking algorithm is described in detail as below:

1) The continuous incoming news texts to be tracked is represented as the double vectors using the previous double vectors text representation model, and the time window based DBSCAN is applied to do the fine-grained topic detection with a higher similarity threshold generating many sub-topic clusters of which the elements are very similar to each other. Specifically, a higher similarity threshold works as the function to remove the noisy data and cluster very identical news texts.
2) Representing the sub-topic as a double vectors text representation which is defined as below:

$$Sub\_title\_vec_i = \frac{1}{n} \times \sum_{i=1}^{n} title\_vec_i, \tag{17}$$

$$Sub\_content\_vec_i = \frac{1}{n} \times \sum_{i=1}^{n} content\_vec_i, \tag{18}$$

where the $Sub\_title\_vec_i$ and $Sub\_content\_vec_i$ are the double vectors text representation of the sub-topic sub_i, and the $title\_vec_i$ and $content\_vec_i$ are the double vectors representation of the news text $text_i$ in sub-topic sub_i. Apparently, each sub-topic is represented as an average of the double vectors representation of the news texts belonging to it. The time feature of the sub-topic is defined as the time interval from the earliest published time to the latest published time of the news texts which belong to the sub-topic. Then distributing the sub-topic clusters to all the slave servers.
3) Taking out the training samples of which the timestamp is within the time interval [$timestamp_0$- T, $timestamp_0$] defined by the time feature of the sub-topic from the prior texts which serves as the training dataset, where T is a time window which is used to reduce the computations, and the $timestamp_0$ is the beginning time of the sub-topic. Computing the similarity between the sub-topic and the training samples.
4) Filtering out the training samples of which the similarity with the sub-topic is above the threshold T_b, and sorting them in descending order according to the computed similarity. If the similarity is below the threshold T_b which indicates that the news texts in this sub-topic do not belong to the existing topics in the training dataset, and the sub-topic with its news text will be discarded.
5) Taking out the Top-K training samples or all the training samples in case there is less than K training samples obtained in step (4), and voting on the topic label of the sub-topic by their topic labels. The most frequent topic label will be the topic label of the news texts in the sub-topic.
6) Finally, these new texts which have been tracked will be added to the prior texts to expand the training dataset for the next iteration.

V. EXPERIMENTS AND EVALUATIONS

In this section, first of all, the experimental environment about the software and hardware and the evaluation metrics have been given. Baseline algorithms to make comparisons with the proposed algorithm in this paper are also introduced in brief. Then experiments about the proposed time window based parallelized topic detection algorithm are conducted to validate its effectiveness and efficiency followed by the analysis of the experimental results. Furthermore, the comparison between the trend development map generated using the overall topic detection and tracking technique integrated with the proposed improved algorithms and the one from the professional topic tracking websites have been made. Finally, the comparison experiments of the proposed improved KNN based topic tracking algorithm and some baseline algo-

TABLE 1. Hardware and software list.

Name	Category	Description
RAM	Hardware	256G
Hard disk capacity	Hardware	2T
Number of cores of CPU	Hardware	56
Operation system	Software	Ubuntu16
Python	Software	Version 3.5
Spark	Software	Version 2.3.9

rithms are performed to validate the improved performance in accuracy.

A. EXPERIMENT SETTINGS

1) HARDWARE AND SOFTWARE

The hardware and software used to implement the topic detection and tracking algorithm are listed in Table 1. Note that the implementation is achieved on the basis of the Spark platform. However, limited by the hardware condition of the laboratory, only a deep learning server is exploited to construct a pseudo-distributed cluster of servers to mimic the parallel distributed computation.

2) DATASET

To collect news texts online efficiently, the data collection layer of TDSBD is designed and employed in a parallelized way. The dataset is made up of the news texts collected from mainstream web portals such as the sina.com, ifeng.com and People's Daily Online and so on. The published time of these news texts ranges from September 28th, 2018 to October 6th, 2018, and the total number of the collected news texts is 44149. Note that the number of news texts belonging to a specified topic can vary extremely, which conforms to the Long Tail Effect in the real world. In terms of the hot topic detection, those which have fewer pieces of news associated with it are less likely to be the hot topics focused and discussed by the media. Hence, in this paper, the cluster which contains fewer pieces of news will be excluded in the topic detection tasks to improve the efficiency of the algorithms. The distribution of the dataset is shown in table 2 in which only the topics containing more news texts are displayed.

The dataset is divided into two datasets for different tasks and purposes, and the descriptions of these two datasets are listed below:

Dataset1: The labeled topic news texts are chosen from the whole dataset, and will be used for topic detection and tracking under no disturbance of the noise data.

Dataset2: All the original news texts with unlabeled topic news texts are included in Dataset2, and this dataset is used for the topic detection and tracking under the existence of noise data. Meanwhile, it is also exploited in the hot topic detection.

Since not all words in the news texts are helpful in distinguishing one news text from another which may not share the same topic, all the news texts in the dataset are processed by stop words removal before taken as input data.

TABLE 2. The distribution of the dataset.

Starting time of the topic	Topic	The number of the news text contained
2018-09-28	Indonesia tsunami	206
2018-10-05	America sanctions against Iran	148
2018-10-06	China International Import Expo	514
2018-10-06	Saudi Arabia admits the death of journalist	623
2018-10-06	The opening of the Hong Kong-Zhuhai-Macao Bridge	163
2018-10-20	Immigrants gather at the US-Mexico Border	216
2018-10-20	America withdraws from the China Missile Treaty	376
2018-10-21	Xi Jinping talks about private enterprises	134
2018-10-24	Mail-Bomb Scare in the US	238
2018-10-28	Chongqing Bus falling into the river	598
2018-10-29	Indonesian passenger airplane crashed into the sea	550
2018-10-30	Li Yong's death	130
2018-10-30	Jin Yong's death	284
2018-11-03	Lanzhou highway toll station crash	130
2018-09-28	Other topics	39839

3) EVALUATION METRICS

To evaluate the accuracy of the topic detection and tracking task, the precision and recall evaluation metrics are utilized. The precision evaluates the percentage of the correct labeled news texts in a specified topic cluster, while the recall evaluates the percentage of the correct labeled news texts with respect to all the news texts in the whole dataset which share the same topic with the labeled news texts. The F_1-Score which is computed by the recall and precision can be used to evaluate both precision and recall conveniently. The bigger F_1-Score is, the more accurate the topic detection and tracking model is. The relationship of the precision, recall and F_1-Score is described as below:

$$F_1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{19}$$

To evaluate the proposed KNN based parallel topic tracking, the trend development map which reflects how the hotness of a topic changes over time generated using the proposed methods is compared with the one generated by the professional topic tracking website.

4) BASELINE ALGORITHMS

The selected baseline algorithms for text representation and for clustering are listed in table 3 and table 4 respectively.

Some combinations of the baseline algorithms for text representation and for clustering are compared with the proposed topic detection algorithm.

The selected baseline algorithms for topic tracking includes KNN and Single-Pass. Some combinations of the

TABLE 3. Baseline algorithms for text representation.

Baseline algorithms for text representation	Description
VSM-TFIDF	The entry based improved VSM with TF-IDF
VSM-TFIDF-POS_Entity	The improved model by adding the location feature and named entities to the VSM-TFIDF model.
word2vec-TFIDF	TF-IDF based word embedding model.
wrod2vec-TFIDF-POS_Entity	The improved model by adding the location feature and named entities to the word2vec-TFIDF model.

TABLE 4. Baseline algorithms for clustering.

Baseline algorithms for clustering	Description
DBSCAN	Traditional density based clustering algorithm.
Single-Pass	One of the widely used clustering algorithms in topic detection.

baseline algorithms for topic tracking and for text representation are compared with the proposed parallelized KNN based topic tracking algorithm.

B. ANALYSIS OF THE TIME WINDOW BASED PARALLELIZED TOPIC DETECTION EXPERIMENTAL RESULTS

1) ACCURACY OF TOPIC DETECTION TASK

Table 5 shows the comparisons of accuracy of the baseline algorithms and proposed algorithm for topic detection on Dataset1 and Dataset2. It can be concluded that the proposed algorithm achieves the best performance from the point of the F_1-Score both on Dataset1 and Dataset2. In terms of the precision and recall, the proposed algorithm for topic detection has achieved 99.9% and 92.8% with regarding to precision on Dataset1 and Dataset2 respectively, and the recall on the Dataset1 and Dataset2 are 94.8% and 75.0%, which has the comparable results with some baseline algorithms under no disturbance of noise data. The proposed algorithm significantly outperforms some baseline algorithms with respect to recall under existence of noise data.

2) EFFICIENCY OF THE TIME WINDOW BASED CLUSTERING

Since the time complexity of the Single-Pass algorithm is equal to the DBSCAN which is $O(n^2)$ at the worst conditions, it can be utilized to make comparisons with the proposed algorithm. For the sake of fairness, the comparison experiment is conducted using the proposed time window based DBASCAN but in serial way and the traditional Single-Pass clustering. Assume that the number of the news texts generated every day is 1000, and let Improved_3 denote the DBSCAN algorithm with a time window of which the size is 3 days, and let Improved_5 denote the DBSCAN algorithm with a time window of which the size is 5 days. Give the input

TABLE 5. Comparisons of accuracy using different topic detection algorithms.

Topic detection algorithms	Dataset1			Dataset2		
	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
VSM-TFIDF	0.997	0.931	0.962	0.882	0.699	0.779
VSM-TFIDF-POS_Entity	0.993	0.937	0.964	0.872	0.746	0.804
word2vec-TFIDF	0.995	0.870	0.928	0.956	0.640	0.766
word2vec-TFIDF-POS_Entity	0.987	0.933	0.959	0.946	0.605	0.738
VSM-TFIDF	1	0.895	0.944	0.817	0.680	0.742
VSM-TFIDF-POS_Entity	0.913	0.798	0.851	0.911	0.576	0.705
word2vec-TFIDF	1	0.843	0.914	0.934	0.513	0.662
word2vec-TFIDF-POS_Entity	1	0.862	0.925	0.924	0.601	0.728
Proposed algorithm	0.999	0.948	0.973	0.928	0.750	0.824

data generated simply by the standard normal distribution, the experimental result is presented as Figure 5.

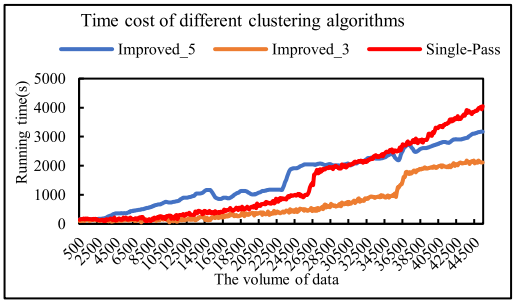

FIGURE 5. The comparisons between different clustering algorithms.

Figure 5. shows that the running time of the proposed time window based DBSCAN is increasing as the size of the time window becomes bigger. It can be concluded that the time cost of the proposed algorithm almost grows linearly with the volume of data getting larger, and the traditional Single-Pass obviously has a quadratic growth as the volume of data increases. Therefore, the proposed algorithm can efficiently reduce the time complexity for topic detection.

C. ANALYSIS OF THE HOT TOPIC DETECTION EXPERIMENTAL RESULTS

1) EFFECTIVENESS OF HOT TOPIC DETECTION TASK

To evaluate the effectiveness of the hot topic detection proposed in this paper, the comparison of the hot topic ranking lists from the proposed hot topic detection system and two professional hot topic detection websites, namely, Memorabilia and Eefung, are made. Each hot topic ranking list includes Top-10 hot topics generated from the Dataset2 using

TABLE 6. Comparison of different hot topic ranking lists.

Method	The proposed method	Memorabilia	Eefung
Top-10 hot topics	Saudi Arabia admits the death of journalist	Saudi Arabia admits the death of journalist	Fan Bingbing evades taxes
	Chongqing Bus falls into the river	Chongqing Bus falls into the river	Alipay Koi
	Indonesian passenger airplane crashed into the sea	Indonesian passenger airplane crashed into the sea	Chongqing Bus falls into the river
	The US withdraws from the China Missile Treaty	Jin Yong's death	Saudi Arabia admits the death of journalist
	China International Import Expo	Escape of the two felons in Lingyuan third prison	China International Import Expo
	The opening of the Hong Kong-Zhuhai-Macao Bridge	Suspected high speed rail molestation by men	Jin Yong's death
	Immigrants gather at the US-Mexico Border	Rockburst accident of Longyun coal industry	Online Red insults National Anthem
	Mail-Bomb Scare in the US	China's first successful amphibious aircraft "Kunlong" on water	Violent injuries in Peking University Hospital
	Jin Yong's death	Xinhua men's suicide insurance fraud	The opening of the Hong Kong-Zhuhai-Macao Bridge
	America sanctions against Iran	The Shanghai index fell below the bottom of the fuse by 2638 points	Child robbing incident in Beijing

the corresponding algorithm. Table 6 shows the comparison result for hot topic detection task.

Table 6 shows that the hot topics ''Saudi Arabia admits the death of journalist'', ''Chongqing Bus falls into the river'' and ''Jin Yong's death'' have been identified as hot topics by all three methods or systems, and the hot topics like ''Indonesian passenger airplane crashed into the sea'' and ''The opening of the Hong Kong-zhuhai-Macao Bridge'' identified as hot topics by the proposed method are partially identified by the other two methods. As a result, the hot topic detection algorithm in this paper can identify some hot topics at a moderate level due to sixty percentage of the intersection with the results from professional websites.

2) ANALYSIS ABOUT TREND DEVELOPMENT MAP OF HOT TOPIC

To validate the effectiveness of the proposed hot topic detection, some trend development maps of hot topics are plotted, which demonstrates how the hotness value of a hot topic changes over a life cycle. There are two curves that are from the proposed topic detection algorithm integrated with the hotness calculation formula and the professional hot topic detection website called 360 Trends respectively in a trend development map for a hot topic. Such two trend development maps about ''the case of Saudi Arabian journalist missing'' and ''Chongqing bus falls into the river'' are presented in Figure 6. and Figure 7. respectively.

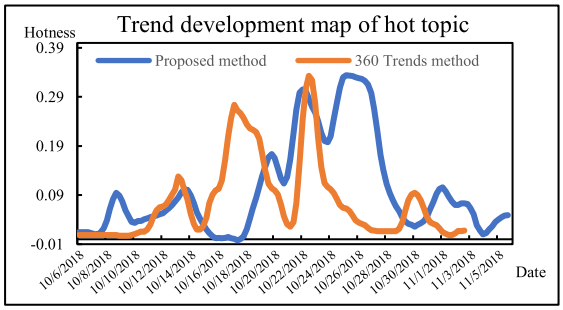

FIGURE 6. Trend development map of the case of Saudi Arabian journalist missing.

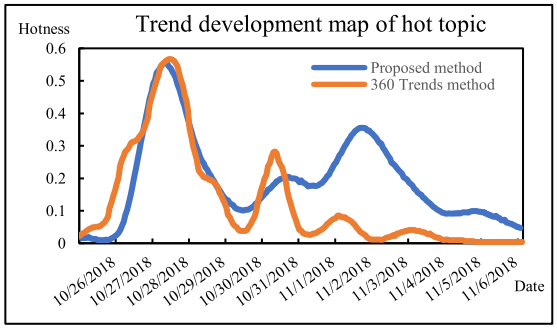

FIGURE 7. Trend development map of the case of Chongqing Bus falls into the river.

D. ANALYSIS OF EXPERIMENTAL RESULTS OF THE IMPROVED KNN BASED TOPIC TRACKING

To validate the accuracy of the improved topic tracking algorithm, the news texts are sorted by its timestamp, then for each topic the top 5% of the news texts are selected as the initial known labeled topic news texts, and the others are used as the test dataset for later topic tracking.

Some parameters in the topic tracking experiment are described as below: the time window T is set to 8 days, the fine-grained similarity threshold T_s is set to 0.9, and the threshold T_b used in the improved KNN algorithm is set to 0.5, and the number of the nearest neighbors K is set to 6. The experimental result is presented in Table 7 below. It can be observed that the average of the F_1-score on above ten tracked topics is 0.978, and the precision of each tracked topic is no less than 0.97, and the recall for every topic except for the topic ''Li Yong's death'' is no less than 0.90.

The comparison experiments are also performed and the results are presented in Table 8. It can be concluded that the recall and F1-score by the proposed improved KNN

TABLE 7. The result of the topic tracking experiment.

Topic	Precision	Recall	F_1-score
Saudi Arabian journalist's death	1	0.99	0.99
Indonesian passenger airplane crashed into the sea	0.98	0.99	0.99
Chongqing Bus falls into the river	0.98	0.99	0.98
The US withdraws from the China Missile Treaty	0.99	0.99	0.99
Indonesia tsunami	1	0.91	0.95
The opening of the Hong Kong-Zhuhai-Macao Bridge	0.99	0.93	0.96
Immigrants gather at the US-Mexico Border	0.97	0.99	0.98
Li Yong's death	1	0.86	0.92
Lanzhou highway toll station crash	1	0.90	0.95
Jin Yong's death	1	0.91	0.95

TABLE 8. Comparison results of different topic tracking algorithms.

Algorithms	Precision	Recall	F1-Score
The proposed improved KNN parallelized algorithm	0.989	0.969	0.978
Double vectors + KNN	0.967	0.962	0.962
Double vectors + Single-Pass	0.990	0.912	0.947
word2vec_average+KNN	0.964	0.959	0.959
word2vec_average+Single-Pass	0.910	0.890	0.889

algorithm are superior to other compared algorithms, the precision is only 0.001 lower than the highest value. Note that the F_1-score of the proposed algorithm has an improvement of 0.016 higher than the highest value among the compared algorithms, and the precision has outperformed the traditional KNN algorithm because the fine-grained similarity threshold is applied to filter news texts with lower similarity to improve the precision of the KNN clustering.

VI. CONCLUSIONS AND FUTURE WORKS

In view of the vast number of news texts generated by the websites on the Internet and the problem of information overload, the news collecting and topic detection framework TDSBD is proposed with implementation on Spark. The improved time window based DBSCAN is proposed to implement the topic detection module, which proves to be effective and efficient with lower time complexity. Hot topic detection has been explored with the comparison of the trend development map of hot topics generated from the proposed method and professional topic tracking websites. The improved KNN based parallelized topic tracking algorithm is proposed to improve the accuracy and efficiency facing with the challenge of big data. Related experiments are conducted to validate the effectiveness and efficiency of the parallelized topic detection system. However, the text representation method used in this paper may fall behind deep learning-based methods or models, and the formula of self-adaptive similarity threshold used in topic detection task is still worth discussing to improve the performance of the clustering algorithm. Moreover, combing the microblog data with the news texts as the data source for hot topic detection will be interesting future work.

REFERENCES

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," presented at the ICLR, 2013. [Online]. Available: https://dblp.org/rec/journals/corr/abs-1301-3781.html
[2] Z. Jiaming, X. Yaoyi, and W. Bo, "Method of micro-blog event tracking based on word vector," *Comput. Eng. Appl.*, vol. 52, no. 17, pp. 73–78, 2016.
[3] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.
[4] X. Li and W. B. Croft, "Statistical language modeling for information retrieval," *Ann. Rev. Inf. Sci. Technol.*, vol. 39, no. 1, pp. 1–31, 2005.
[5] V. Lavrenko, J. Allan, E. DeGuzman, D. L. Flamme, V. Pollard, and S. Thomas, "Relevance models for topic detection and tracking," presented at the Int. Conf. Hum. Lang. Technol. Res., 2002. [Online]. Available: https://wenku.baidu.com/view/c671c048c850ad02de8041d6.html
[6] Z. Peng, G. Huantong, and C. Qingsheng, "An approach of Chinese text representation based on semantic and statistic feature," *J. Chin. Comput. Syst.*, vol. 28, no. 7, pp. 1311–1313, 2007.
[7] J. Peng, D. Q. Yang, S. W. Tang, Y. Fu, and H. Jiang, "A novel text clustering algorithm based on inner product space model of semantic," *Chin. J. Comput.*, vol. 30, no. 8, pp. 1354–1363, 2007.
[8] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1998, vol. 51, no. 2, pp. 37–45.
[9] N. Luo, W. L. Zuo, F. Y. Yuan, J. B. Zhang, and H. J. Zhang, "Using ontology semantics to improve text documents clustering," (English Edition), *J. Southeast Univ.*, vol. 22, no. 3, pp. 370–374, 2006.
[10] L. Tao, W. Gongyi, and C. Zheng, "An effective unsupervised feature selection method for text clustering," *J. Comput. Res. Develop.*, vol. 42, no. 3, pp. 381–386,2005.
[11] 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 Congress)*, San Francisco, CA, USA. Jun. 2016, pp. 251–258, doi: 10.1109/BigDataCongress.2016.39.
[12] Q. Zhou, L. Shi, L. Xu, and W. Liu, "An improved single-pass topic detection method," in *Proc. 10th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, Changsha, China, Feb. 2018, pp. 317–320.
[13] M. Sun and C. Liu, "Research on hot topic detection based on DBSCAN algorithm and inter sentence relationship," *Library Inf. Service*, vol. 61, no. 12, pp. 113–121, 2017.
[14] J. Carbonell, Y. Yang, J. Lafferty, R. D. Brown, T. Pierce, and X. Liu, "CMU report on TDT-2: Segmentation, detection, and tracking," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA. Accessed: Dec. 26, 2020. [Online]. Available: http://www.cs.cmu.edu/~jgc/publication/CMU_Approach_TDT_Segmentation_DARPA_1999.pdf
[15] C. Linjun, "Research of topic detection and tracking based on multisource data," M.S. thesis, Dept. Comput. Sci. Eng., Univ. Electron. Sci. Technol. China, Chengdu, China, 2017.
[16] W. Xinxing, "The research on parallelization of Spark based hot topic detection algorithm," *Softw. Guide*, vol. 15, no. 9, pp. 51–54, 2016.
[17] W. Ai and D. Li, "Parallelizing hot topic detection of microblog on spark," in *Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Changsha, China, Aug. 2016, pp. 1461–1468, doi: 10.1109/FSKD.2016.7603392.
[18] C. Lei-Lei, "Text clustering study with K-means algorithm of different distance measures," *Software*, vol. 36, no. 1, pp. 56–61, 2015.
[19] T. Shixiao, D. Lixin, and Z. Jinqiu, "K-means text clustering algorithm based on density peaks," *Comput. Eng. Des.*, vol. 38, no. 4, pp. 1019–1023, 2017.
[20] D. H. Zhai, Y. Jiang, F. Gao, Y. U. Lei, and F. Ding, "K-means text clustering algorithm based on initial cluster centers selection according to maximum distance," *Appl. Res. Comput.*, vol. 31, no. 3, pp. 713–715, 2014.
[21] Y. Zhao, K. Zhang, H. Zhang, X. Yan, and Y. Cai, "Hot topic detection based on combined content and time similarity," in *Proc. Int. Conf. Prog. Informat. Comput. (PIC)*, Nanjing, China, Dec. 2017, pp. 399–403.
[22] G. Salton and C. T. Yu, "On the construction of effective vocabularies for information retrieval," *ACM SIGPLAN Notices*, vol. 10, no. 1, pp. 48–60, 1975.
[23] R. Nallapat, "Semantic language models for topic detection and tracking," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol., HLT-NAACL Student Res. Workshop-Volume Assoc. Comput. Linguistics*, 2003, pp. 1–6.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 3111–3119.
[25] M. Franz, T. Ward, J. S. McCarley, and W.-J. Zhu, "Unsupervised and supervised clustering for topic tracking," in *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New Orleans, LA, USA, 2001, pp. 310–317.
[26] W. Qiong, "An improved K-means optimization approach for text clustering," *Comput. Modernization*, no. 3, pp. 48–51, 2015. [Online]. Available: http://www.c-a-m.org.cn/CN/10.3969/j.issn.1006-2475.2015.03.010
[27] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2001, pp. 97–106.
[28] Y. Haibo, "Research on the key technology of hot spot topic discovery based on microblogging," M.S. thesis, School Electron. Inf. Eng., Lanzhou Jiaotong Univ., Lanzhou, China, 2017.
[29] H. Haiping, "Emerging topic detection from social media," M.S. thesis, School Math. Comput. Sci., Fuzhou Univ., Fuzhou, China, 2017.
[30] X. Yang, "Design and implementation of the micro-blog topic detection system based on incremental clustering," M.S. thesis, School Data Comput. Sci., Sun Yat-sen Univ., Guangzhou, China, 2012.
[31] S. Mingxi and L. Chunqi, "The research on hot topic detection based on DBSCAN and sentence relationship," *Library Inf. Service*, vol. 61, no. 12, pp. 113–121, 2017.
[32] K. Rajaraman and T. Ah-Hwee, "Topic detection, tracking, and trend analysis using self-organizing neural networks," in *Proc. PAKDD*, Hong Kong, 2001, pp. 102–107.
[33] H. Diao, Z. Bai, and X. Yu, "Notice of retraction: The application of improved K-nearest neighbor classification in topic tracking," in *Proc. Int. Conf. Educ. Inf. Technol.*, Chongqing, China, Sep. 2010, pp. V2-64–V2-68, doi: 10.1109/ICEIT.2010.5607527.
[34] H. Zhang, J. Zhou, L. Wang, and L. Zhao, "An adaptive topic tracking model based on 3-dimension document vector," *J. Chin. Inf. Process.*, vol. 5, no. 24, pp. 70–76, 2010.
[35] X. Liu, F. Ren, and C. Yuan, "Use relative weight to improve the kNN for unbalanced text category," in *Proc. 6th Int. Conf. Natural Lang. Process. Knowl. Eng. (NLPKE)*, Beijing, China, Aug. 2010, pp. 1–5.
[36] S. Arora, L. Yingyu, and M. Tengyu, "A simple but tough-to-beat baseline for sentence embeddings," presented at the ICLR, 2017. [Online]. Available: https://openreview.net/forum?id=SyK00v5xx
[37] Y. Hui-Min, C. Wei, and D. Guan-zhong, "Design and implementation of on-line hot topic discovery model," *Wuhan Univ. J. Natural Sci.*, vol. 11, no. 1, pp. 21–26, Jan. 2006.
[38] (2019). *China Webmaster*. [Online]. Available: https://top.chinaz.com/hangye/index_zonghe_menhu.html

CHUANZHEN LI received the Ph.D. degree from the Communication University of China, in 2013. She is currently an Associate Professor with the Digital Media Technology Department, Communication University of China. Her research interests include media data analysis, recommendation algorithm, and multimodal information processing.

MINQIAO LIU received the B.S. degree in communication engineering from the Communication University of China, in 2019, where he is currently pursuing the M.S. degree in information and communication engineering. His research interests include information collection and analysis, recommendation systems, and artificial intelligence.

JUANJUAN CAI received the M.S. degree from the Communication University of China, in 2007. She is currently an Associate Researcher with the Key Laboratory of Media Audio and Video, Ministry of Education, Communication University of China. Her research interests include multimedia technology, audio signal processing, and big data analysis.

YANG YU received the M.S. degree from the Communication University of China, in 2019. He is currently with iQIYI Inc., Beijing, China. His research interests include natural language processing, hot topic detection, text style transfer, multimodal fusion model, and word sense disambiguation.

HUI WANG received the Ph.D. degree from the Communication University of China, Beijing, China, in 2011. He is currently a Professor with the State Key Laboratory of Media Convergence and Communication, Communication University of China. His research interests include audio signal processing, media convergence, sound field reproduction and broadcasting, and television technology.