

## Research Article

# A Novel Text Clustering Approach Using Deep-Learning Vocabulary Network

Junkai Yi,<sup>1,2</sup> Yacong Zhang,<sup>1</sup> Xianghui Zhao,<sup>2</sup> and Jing Wan<sup>1</sup>

<sup>1</sup>College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

<sup>2</sup>China Information Technology Security Evaluation Center, Beijing 100085, China

Correspondence should be addressed to Xianghui Zhao; zxhitsec@sina.com

Received 9 October 2016; Revised 1 February 2017; Accepted 16 February 2017; Published 15 March 2017

Academic Editor: Nazrul Islam

Copyright © 2017 Junkai Yi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Text clustering is an effective approach to collect and organize text documents into meaningful groups for mining valuable information on the Internet. However, there exist some issues to tackle such as feature extraction and data dimension reduction. To overcome these problems, we present a novel approach named deep-learning vocabulary network. The vocabulary network is constructed based on related-word set, which contains the “cooccurrence” relations of words or terms. We replace term frequency in feature vectors with the “importance” of words in terms of vocabulary network and PageRank, which can generate more precise feature vectors to represent the meaning of text clustering. Furthermore, sparse-group deep belief network is proposed to reduce the dimensionality of feature vectors, and we introduce coverage rate for similarity measure in Single-Pass clustering. To verify the effectiveness of our work, we compare the approach to the representative algorithms, and experimental results show that feature vectors in terms of deep-learning vocabulary network have better clustering performance.

## 1. Introduction

Webpages, microblogs, and social networks provide much useful information for us, and text clustering is an important text mining method to collect valuable information on the Internet. Text clustering helps us to group an enormous amount of text documents into small meaningful clusters, which have been used in many research fields such as sentiment analysis (opinion mining) [1–3], text classification [4–6], text summarization [7], and event tracking and topic detection [8–10].

The process of text clustering is usually divided into two phases: preprocessing phase and clustering phase. Before preprocessing phase, there are some basic steps (including tokenization, remove-stop-words, and stemming-word) needed to process text documents, and these steps split sentences into words and remove useless words or terms.

The first phase is the preprocessing of text, and the second phase is clustering for text documents. The preprocessing phase is mainly to transform text documents into structured data that can be processed by clustering algorithms. This

phase contains two parts: feature extraction and feature selection.

In existing scientific literatures, there are two categories of feature extraction methods: term frequency-based method and semantic web-based method. Term frequency-based method is a method of counting words' number, and semantic web is to construct the knowledge in certain domain to an ontology, which contains words and their relations.

Term-document vectors are extracted from text documents in the process of feature extraction. Most term frequency-based methods employ vector space model (VSM) to represent text documents, and each entry of VSM is the frequency of words or terms. The most representative method based on term frequency is term frequency-inverse document frequency (tf-idf) algorithm. For its simplicity and high efficiency, researchers have proposed many improved tf-idf algorithms [11, 12].

However, the relations of words (or word order) are lost when text documents are transformed into term-document vectors. Many researchers find that the words or terms have lexical “cooccurrence” phenomenon [13], which means some

words or terms have a high probability of occurrence in a text document. Researchers think that the “cooccurrence” relations of words or terms can generate more precise feature vectors to represent the meaning of text documents.

The objective of feature selection is to remove redundant information and reduce the dimensionality of term-document vectors. The methods of feature selection are categorized as corpus-based method, Latent Semantic Indexing (LSI), and subspace-based clustering. The corpus-based method merges synonyms together to reduce the dimensionality of features, which depends on large corpora such as WordNet and HowNet. Traditional LSI decomposes a term-document vector into a term-space matrix by singular value decomposition (SVD). Subspace-based clustering groups text documents in a low-dimensional subspace.

In our paper, we propose a novel approach to address two issues: one is the loss of word relations in the process of feature extraction, and the other is to retain the word relations in dimension reduction. Considering that the relations of words and terms are lost in term frequency-based methods, we construct a vocabulary network to retain “cooccurrence” relations of words or terms. Term frequency is replaced with the “importance” of words or terms in VSM. Furthermore, traditional feature selection methods can lose some information that affects the performance of clustering [14], and we introduce deep learning for dimension reduction.

The main contributions of our paper are that we present a novel graph-based approach for text clustering, called deep-learning vocabulary network (DLVN). We employ the edges of vocabulary network to represent the relations between words or terms and extract features of text documents in terms of related-word set. The related-word set is a set of words in the same class, and we utilize association rules learning to obtain relations between words. In addition, high dimensional and sparse features of text have a big influence on clustering algorithms, and we employ deep learning for dimensionality reduction. Accordingly, an improved deep-learning Single-Pass (DL-SP) is used in the process of clustering. To verify the effectiveness of the approach, we provide our experimental evaluation based on Chinese corpora.

The rest of this paper is organized as follows. Section 2 reviews related work in previous literatures. Section 3 introduces theoretical foundation related to this paper. Section 4 describes the approach of DLVN we propose. Section 5 is experimental analysis. Section 6 is the conclusion of our work.

## 2. Related Work

Text clustering groups text documents of similar content (so-called topic) into a cluster. In this section, we use three subsections to review related literatures.

**2.1. Feature Extraction.** Term frequency-based method is an important method to extract features. In term frequency-based method, text documents are represented as VSM, and each document is transformed into a vector, whose entries are the frequency of words or terms. Most term frequency-based methods are to improve tf-idf.

Semantic web is to structure knowledge into an ontology. As researchers find that the relations between words contribute to understanding the meaning of text, they construct a semantic network in terms of concepts, events, and their relations. Yue et al. [15] constructed a domain-specific ontology to describe the hazards related to dairy products and translated the term-document vectors (namely, feature vectors of text) into a concept space. Wei et al. [16] exploited an ontology hierarchical structure for word sense disambiguation to assess similarity of words. The experiment results showed better clustering performance for ontology-based methods considering the semantic relations between words. Bing et al. [17] proposed an adaptive concept resolution (ACR) model for the characteristics of text documents, and ACR was an ontology-based method of text representation. However, the efficiency of semantic web analysis is a challenge for researchers, and the large scale of text corpora has a great influence on algorithms [18].

For retaining the relations of words and terms, some researchers proposed to employ graph-based model in text clustering [19, 20]. Mousavi et al. [21] proposed a weighted-graph representation of text to extract semantic relations in terms of parse trees of sentences. In our work, we introduce frequent itemsets to construct related-word set, and use each itemset of related-word set to represent the relations between words. Language is always changing, and new words are appearing every day. Related-word set can capture the change of language by mining frequent itemsets.

**2.2. Feature Selection.** Feature selection is a feature construction method to transform a high dimensional feature space into a low-dimensional feature space. SVD is a representative method using mathematical theory for dimension reduction. Jun et al. [22] combined SVD and principal-component analysis (PCA) for dimensionality reduction. Zhu and Allen [23] proposed a latent semantic indexing subspace signature model (LSISMM) based on LSI and transformed term-document vectors into a low-rank approximation for dimensionality reduction. However, LSI selects a new feature subset to construct a semantic space, which loses some important features and suffers from the irrelevant features.

Due to the sparsity and high-dimensionality of text features, the performance of the subspace-based clustering is better than traditional clustering algorithm [24, 25]. Moreover, some researchers integrate many related theories for dimensionality reduction. Bharti and Singh [26] proposed a hybrid intelligent algorithm, which integrated binary particle swarm optimization, chaotic map, dynamic inertia weight, and mutation for feature selection.

**2.3. Clustering Algorithm.** Clustering is an unsupervised approach of machine learning, and it groups similar objects into a cluster. The most representative clustering algorithm is partitional clustering such as *k-means* and *k-medoids* [27], and each cluster has a center called centroid in partitional clustering. Mei and Chen [28] proposed a clustering around weighted prototypes (CAWP) based on new cluster representation method, where each cluster was represented by multiple objects with various weights. Tunali et al. [29] improved

spherical *k-means* (SKM) and proposed a multicluster spherical *k-means* (MCSKM), which allowed documents to be assigned more than one cluster. Li et al. [30] introduced a concept of neighbor and proposed a parallel *k-means* based on neighbors (PKBN).

Another representative clustering algorithm is hierarchical clustering, which contains divisive hierarchical clustering and agglomerative hierarchical clustering [31]. Peng and Liu [32] proposed an incremental hierarchical text clustering approach, which represented a cluster hierarchy using CF-tree. In addition, Chen et al. [33] proposed an improved density clustering algorithm named density-based spatial clustering of applications with noise (DBSCAN). DBSCAN was sensitive to choosing parameters; the authors combined *k-means* to estimate the parameters.

Ensemble clustering is another clustering algorithm. Ensemble clustering combines the multiple results of different clustering algorithms to obtain final results. Multiview clustering is an extension of ensemble clustering and combines different data that have different properties and views [34, 35].

Matrix factorization-based clustering is an important clustering approach [36]. Lu et al. [37] proposed a semisupervised concept factorization (SSCF), which contained nonnegative matrix factorization and concept factorization for text clustering. SSCF integrated penalized and reward terms by pairwise constraints *must-link* constraints  $C_{ML}$  and *cannot-link* constraints  $C_{CL}$ , which implied two documents belonging to the same cluster or different clusters.

Topic-based text clustering is an effective text clustering approach, in which text documents are projected into a topic space. Latent Dirichlet allocation (LDA) is a common topic model. Yau et al. [38] separated scientific publications into several clusters based on LDA. Ma et al. [39] employed the topic model of LDA to represent the centroids of clusters and combined *k-means++* algorithm for document clustering.

In some literatures, additional information is introduced for text clustering such as side-information [40] and privileged information [41]. What is more, several global optimization algorithms are utilized for text clustering such as particle swarm optimization (PSO) algorithm [42, 43] and bee colony optimization (BCO) algorithm [44, 45].

Similarity measure is also an important issue in text clustering algorithms. To compute the similarity between a text document and a cluster is a fundamental problem in clustering algorithms. The most common similarity measure is distance metric such as Euclidean distance, Cosine distance, and Generalized Mahalanobis distance [46]. There exist other similarity measure methods such as IT-Sim (an information-theoretic measure) [47]. Besides similarity measure, measurement of discrimination information (MDI) is an opposite concept to compute the relations of text documents [48–50].

### 3. Theoretical Foundation

In this section, we describe some theories related to our work. This section contains three subsections, which are frequent pattern maximal (FPMAX), PageRank, and deep belief network (DBN).

```

Procedure FPMAX( $T$ )
Input:  $T$  (an FP-tree)
Global:
    MFIT: an MFI-tree
    Head: a linked list of items
Output: The MFIT that contains all MFI's
Method:
(1) if  $T$  only contains a single path  $P$ 
(2) insert Head  $\cup P$  into MFIT;
(3) else for each  $i$  in Header-table of  $T$ 
(4) append  $i$  to Head;
(5) construct the Head-pattern base;
(6) Tail = {frequent items in base};
(7) subset_checking(Head  $\cup$  Tail);
(8) if Head  $\cup$  Tail is not in MFIT;
(9) construct the FP-tree  $T_{Head}$ ;
(10) call FPMAX( $T_{Head}$ );
(11) remove  $i$  from Head.

```

ALGORITHM 1: FPMAX.

**3.1. FPMAX.** FPMAX is a depth-first and recursive algorithm for mining maximal frequent itemsets (MFIs) in given dataset [51]. Before FPMAX is called, frequent pattern tree (FP-tree) is structured to store frequent itemsets, and each branch of the FP-tree is a representation of a frequent itemset. FP-tree includes a linked list head, which contains all items of the dataset. Maximal frequent itemset tree (MFI-tree) is introduced to store all MFIs in FPMAX. The procedure of FPMAX is described Algorithm 1.

**3.2. PageRank.** PageRank is a link-based ranking algorithm, which is used in the Google search engine. Most of webpages on the Internet are connected with hyperlinks, which carry important information. Hence, some webpages pointed by many webpages are considered to include quality information.

Webpages and hyperlinks in PageRank are structured to directed graph  $G = (V, E)$ , where  $V$  is the set of webpages and  $E$  is the set of hyperlinks. Let  $n$  be the total number of webpages. The PageRank score of the webpage  $i$  is defined by

$$P(i) = \sum_{(i,j) \in E} \frac{P(j)}{O_j}, \quad (1)$$

where  $O_j$  is the number of page  $j$  pointing out to other webpages. Let  $P$  be a vector to represent all PageRank scores

$$P = (P(1), P(2), \dots, P(n))^T. \quad (2)$$

Let  $A$  be the adjacency matrix of the graph  $G$  with

$$A_{ij} = \begin{cases} \frac{1}{O_i}, & \text{if } (i, j) \in E \\ 0, & \text{if } (i, j) \notin E. \end{cases} \quad (3)$$

Hence, (1) can be written as the system of equations with

$$P = A^T P. \quad (4)$$

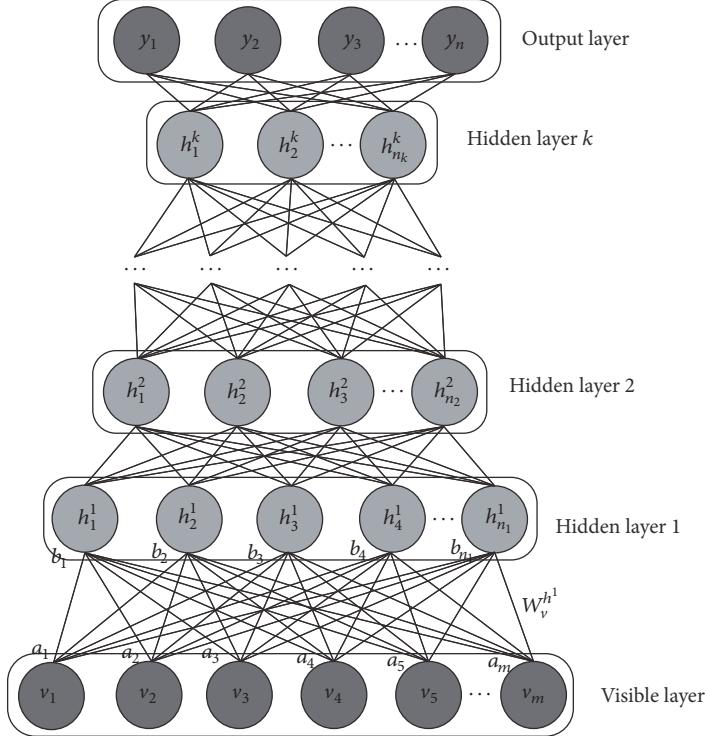


FIGURE 1: The structure of DBN.

PageRank models web surfing as a stochastic process, and the theory of Markov chain can be applied. However, the web graph does not meet the conditions of stochastic process, which requires  $A$  to be stochastic, irreducible, and aperiodic. After the adjustment of  $A$  to fix this problem, we obtain an improved model with

$$P = \left( (1 - d) \frac{E}{n} + dA^T \right) P, \quad (5)$$

where  $E$  is  $ee^T$  ( $e$  is a column vector of all 1's) and thus  $E$  is an  $n \times n$  matrix with all 1's, and  $d$  is a parameter called *damping factor*. After scaling, we obtain

$$P = (1 - d)e + dA^T P. \quad (6)$$

Equation (6) is also transformed as follows:

$$P(i) = (1 - d) + d \sum_{(i,j \in E)} \frac{P(j)}{O_j}. \quad (7)$$

The computation of PageRank score is a process of iteration. Given an initial value of  $P$ , the iteration ends when the score of PageRank does not change or the change is less than a threshold.

**3.3. Deep Belief Network (DBN).** DBN is a model of deep learning and composed of multilayer restricted Boltzmann machines (RBMs). DBN contains the input layer (visible layer), the hidden layers, and the output layer. There are connections between a layer and adjacent layer, but no

connections among units in each layer. The structure of DBN is shown in Figure 1.

As shown in Figure 1, an RBM consists of two adjacent layers. The training of DBN includes two steps, pretraining and fine-tuning. RBM contains a visible layer  $v^{(k)}$  and a hidden layer  $h^{(k)}$ . The parameters of RBM are  $(W^{(k)}, a^{(k)}, b^{(k)})$ .  $(W^{(k)})$  are the weights of connections between the visible layer and the hidden layer, and  $(a^{(k)}, b^{(k)})$  are the bias vectors of the visible units and the hidden units. Giving an initial value to  $W^{(k)}$ , the parameters are updated with

$$W_{ij}^{(k)} = W_{ij}^{(k)} + \eta \nabla W_{ij}^{(k)}, \quad (8)$$

where  $\eta$  is learning rate, and  $(a^{(k)}, b^{(k)})$  are similar to  $W^{(k)}$ .

The gradient of  $\nabla W^{(k)}$  is obtained by Gibbs Sampling.

$$\nabla W_{ij}^{(k)} = E[v_i^{(k)}h_j^{(k)}]_{\text{data}} - E[v_i^{(k)}h_j^{(k)}]_{\text{Gibbs}}, \quad (9)$$

where  $E[\cdot]_{\text{data}}$  and  $E[\cdot]_{\text{Gibbs}}$  are the expectations of data samples and samples from Gibbs Sampling, and  $(\nabla a^{(k)}, \nabla b^{(k)})$  are similar to  $\nabla W_{ij}^{(k)}$ .

DBN is fine-tuned with a set of labeled inputs in terms of error back propagation after the pretraining of DBN. The parameters are updated by

$$W_{ij}^{(k)} = W_{ij}^{(k)} + \eta \nabla W_{ij}^{(k)}, \quad (10)$$

where  $\nabla W_{ij}^{(k)} = h_i^{(k-1)}\delta_j^{(k)}$ , and  $\delta_j^{(k)}$  is an error vector.

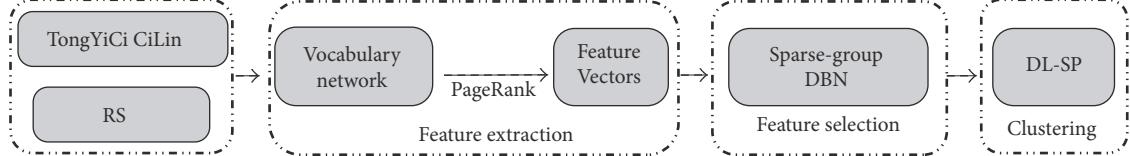
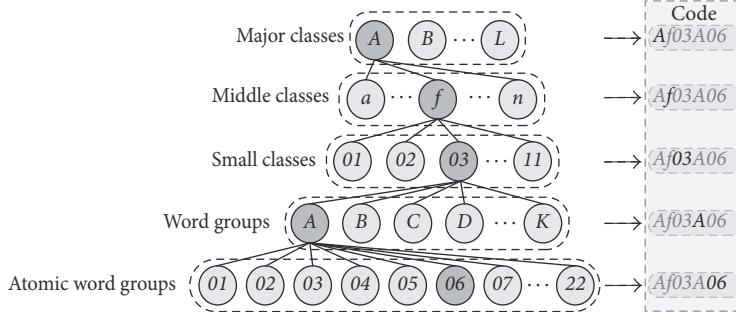


FIGURE 2: The procedure of DLVN.

FIGURE 3: The structure of *TongYiCi CiLin*.

## 4. Deep-Learning Vocabulary Network

In this section, we propose an approach called deep-learning vocabulary network (DLVN) for text clustering. The first step of DLVN is the construction of vocabulary network. The cooccurrence of words or terms is useful information for text clustering. We use the nodes of the vocabulary network to represent words or terms and the edges of the vocabulary network to represent the relations between words or terms. In our work, there are two methods to obtain the cooccurrence relations of words: related-word set and *TongYiCi CiLin*. Frequent itemsets are used to discover the relations of items in database. We create related-word set by frequent itemsets, and each itemset of related-word set is a set of words with cooccurrence relation. PageRank is employed to obtain the “importance” of nodes (feature vectors) instead of the term frequency in VSM. Then, an improved DBN (called sparse-group DBN) is proposed for dimensionality reduction. In the process of clustering algorithm, we present DL-SP for clustering, in which coverage rate is used for similarity measure. The procedure of DLVN is shown in Figure 2.

**4.1. Related-Word Set.** The relations of words or terms are important information in text documents. Usually, natural language has the fixed collocation and corresponding contexts, which means some words or terms have a high probability of occurrence in a text document. Thus, the relations between words are important to represent the meaning of text documents. In our paper, we use frequent itemsets to obtain cooccurrence relations between words or terms.

**Definition 1** (related-word set). Let  $D = \{\text{word}_1, \text{word}_2, \dots, \text{word}_n\}$  be the words of text documents from the same topic and  $\text{sup}[\cdot]$  be the support of itemsets. Given a minimum support  $\text{sup}_{ms}$ ,  $X = \{\text{word}_i, \text{word}_j, \dots, \text{word}_k\}$  is defined as an itemset of related-word set, where  $\text{sup}[X] > \text{sup}_{ms}$ .

FPMAX is a depth-first and recursive algorithm for mining MFIs, and it is based on FP-tree to store frequent itemsets. When a database has a large scale, all itemsets of MFI-tree are detected in subset checking of FPMAX, which has a big influence on the efficiency of FPMAX. For improving the efficiency of FPMAX, we use *TongYiCi CiLin* and string match to compress the FP-tree.

*TongYiCi CiLin* is a Chinese semantic dictionary of synonyms and related words, which organizes all words as a five-layer hierarchical tree. It contains 77,343 words, which are divided into 12 major classes, 94 middle classes, and 1438 small classes. The fourth layer and the fifth layer are further divided into word groups and atomic word groups. We use Figure 3 to illustrate the structure of *TongYiCi CiLin*.

*TongYiCi CiLin* maps an atomic word group into a code: the first layer and the fourth layer are capital letters, the second layer is a lowercase letter, and the third layer and the fifth layer are integers. For example, code “Aa01A02” stands for the atomic word group {man, mankind, human}. We replace the words or terms with the code of word groups in MFI’s mining, which contains 4223 nodes. We randomly select 10 documents from the same topic, and the frequent items (words) are listed in Table 1. As some words belong to the same word group, the number of words is compressed largely.

The structures of FP-trees that are created based on words and word groups are shown in Figure 4. Figure 4(a) is FP-tree of words, and FP-tree of word groups is shown in Figure 4(b). The nodes of FP-tree based on the word groups are fewer than the nodes of FP-tree based on the words.

The MFIs have redundant items in Figure 4(b). For example, the MFIs of Figure 4(b) are listed in Table 2.

MFIs include two categories of word groups in Table 2. The word groups {**a**(Bo21A01), **b**(Bo01A06), **d**(Dd14B36)}, and {**a**(Bo21A01), **c**(Bo21A27)} are closely related, and the word groups {**k**(Da21D01), **i**(Dm04A01), **f**(Cb08A01)} and

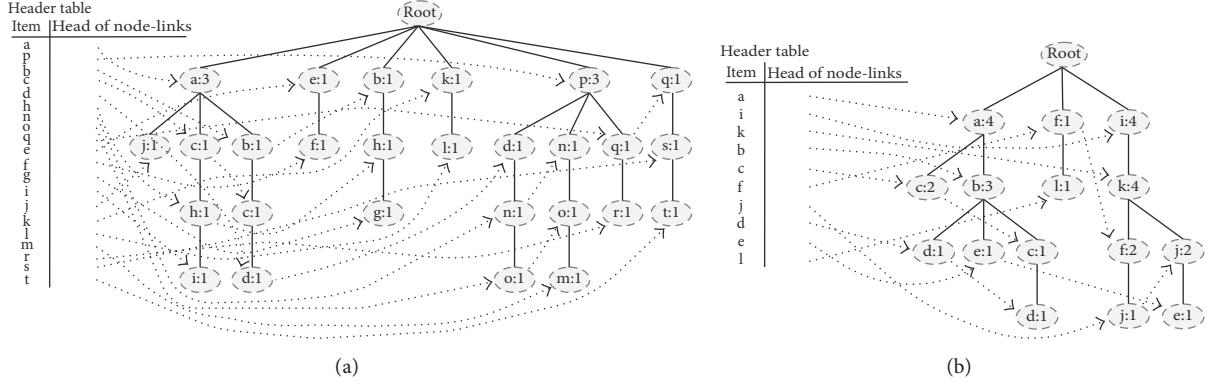


FIGURE 4: The structures of FP-trees.

TABLE 1: The comparison of words and word groups.

Words	Word groups
<b>a</b> (vehicle) <b>b</b> (car) <b>c</b> (engine) <b>d</b> (quality)	<b>a</b> (Bo21A01) <b>b</b> (Bo01A06) <b>e</b> (Dd12A01)
<b>e</b> (automobile) <b>f</b> (truck)	<b>a</b> (Bo21A01) <b>c</b> (Bo21A27)
<b>b</b> (car) <b>g</b> (engine) <b>h</b> (power)	<b>a</b> (Bo21A01) <b>b</b> (Bo01A06) <b>d</b> (Dd14B36)
<b>a</b> (vehicle) <b>i</b> (lorry) <b>c</b> (engine) <b>h</b> (power)	<b>a</b> (Bo21A01) <b>b</b> (Bo01A06) <b>c</b> (Bo21A27) <b>d</b> (Dd14B36)
<b>a</b> (vehicle) <b>j</b> (jeep)	<b>a</b> (Bo21A01) <b>c</b> (Bo21A27)
<b>k</b> (ground) <b>l</b> (situation)	<b>f</b> (Cb08A01) <b>l</b> (Da21A07)
<b>m</b> (area) <b>n</b> (store) <b>o</b> (construction) <b>p</b> (environment)	<b>f</b> (Cb08A01) <b>i</b> (Dm04A01) <b>j</b> (Bn01A01) <b>k</b> (Da21D01)
<b>q</b> (shop) <b>r</b> (building) <b>p</b> (environment)	<b>i</b> (Dm04A01) <b>k</b> (Da21D01) <b>j</b> (Bn01A01)
<b>s</b> (location) <b>q</b> (shop) <b>t</b> (condition)	<b>f</b> (Cb08A01) <b>i</b> (Dm04A01) <b>k</b> (Da21D01)
<b>d</b> (quality) <b>n</b> (store) <b>o</b> (construction) <b>p</b> (environment)	<b>e</b> (Dd12A01) <b>i</b> (Dm04A01) <b>j</b> (Bn01A01) <b>k</b> (Da21D01)

TABLE 2: MFIs of FP-tree based on word groups.

	MFIs
1	<b>a</b> (Bo21A01) <b>b</b> (Bo01A06) <b>d</b> (Dd14B36)
2	<b>a</b> (Bo21A01) <b>c</b> (Bo21A27)
3	<b>k</b> (Da21D01) <b>i</b> (Dm04A01) <b>f</b> (Cb08A01)
4	<b>k</b> (Da21D01) <b>i</b> (Dm04A01) <b>j</b> (Bn01A01)

{**k**(Da21D01), **i**(Dm04A01), **j**(Bn01A01)} are closely related. In fact, the aim of related-word set is to mine the “cooccurrence” of words, and we assume that the relations of words have transitivity. Therefore, we utilize string matching and the same items to combine MFIs.

*Definition 2* (combination of MFIs). Let  $\text{MFIS} = \{\text{MFI}_1, \text{MFI}_2, \dots, \text{MFI}_m\}$  be the MFI’s set obtained from text documents and  $\text{cov}(\cdot)$  be the number of the same items in two MFIs. Suppose that  $\text{cov}(\text{MFI}_1, \text{MFI}_2) > \text{cov}_{\min}$ , where  $\text{cov}_{\min}$  is minimum number of the same items.  $\text{MFI}_1$  and  $\text{MFI}_2$  are removed from  $\text{MFIS}$ , and the combination of  $\text{MFI}_1 \cup \text{MFI}_2$  is add to  $\text{MFIS}$ .

MFIs are inserted into MFI-tree in terms of  $\text{cov}_{\min}$ . For example, given  $\text{MFI}_1 = \{a, b, c, d, e, f\}$ ,  $\text{MFI}_2 = \{a, b, c, d, e, h\}$ ,  $\text{MFI}_3 = \{e, f, g, h, i, j, k\}$ , and  $\text{cov}_{\min} = 0.7$ , the combination of MFIs is  $\text{MFI}_1 \cup \text{MFI}_2 = \{a, b, c, d, e, f, h\}$ . The new

MFI-tree only has two paths ( $\text{MFI}_1 \cup \text{MFI}_2 = \{a, b, c, d, e, f, h\}$  and  $\text{MFI}_3 = \{e, f, g, h, i, j, k\}$ ). The scale of MFI-tree is simplified, and we integrate FPMAX with combination of MFIs to propose an algorithm named FPMAX with related-word set (FPMAX-RS). The step of FPMAX-RS is listed in Algorithm 2.

**4.2. The Construction of Vocabulary Network.** In this section, vocabulary network is constructed to represent text documents, and the vocabulary network contains the relations between words or terms. We employ the “importance” of nodes instead of term frequency in VSM.

**4.2.1. The Selection of Vocabulary Network Nodes.** The word groups in *TongYiCi CiLin* are used as nodes instead of words in vocabulary network. The number of word groups is much fewer than the number of words. In addition, we choose the word groups whose frequency is higher than specified minimal frequency  $f_{\min}$ .

**4.2.2. The Construction of Edges in the Vocabulary Network.** Edges of complex network are the important carrier of information, and the edges of the vocabulary network are used in calculating the “importance” of nodes. Considering the semantic and related information among words of terms, an edge is add to the vocabulary network in terms of the similarity of nodes. Therefore, we add an edge to the vocabulary network if word groups have a closer position

```

Procedure FP MAX-RS( $T$ )
Input:  $T$  (an FP-tree),  $\text{cov}_{\min}$ 
Global:
  MFIT: an MFI-tree
  Head: a linked list of items
Output: The MFIT that contains all MFI's
Method:
(1) if  $T$  only contains a single path  $P$ 
(2)   if  $\text{cov}(\text{Head} \cup P, \text{MFI}) > \text{cov}_{\min}$ 
(3)     combine MFI-tree to this path;
(4)   else
(5)     insert Head  $\cup P$  into MFIT;
(6) else for each  $i$  in Header-table of  $T$ 
(7)   append  $i$  to Head;
(8)   construct the Head-pattern base;
(9)   Tail = {frequent items in base};
(10)  subset_checking (Head  $\cup$  Tail);
(11)  if Head  $\cup$  Tail is not in MFI-tree
(12)    construct the FP-tree  $T_{\text{Head}}$ ;
(13)    call FP MAX-RS( $T_{\text{Head}}$ );
(14)    remove  $i$  from Head.

```

ALGORITHM 2: FP MAX-RS.

in *TongYiCi CiLin*. The semantic similarity of word groups  $\text{sim}(i, j)$  is defined as

$$\text{sim}(i, j) = \frac{\text{depth}(i, j)}{l} \times \frac{\text{TN} - \text{Dis}(i, j) + 1}{\text{TN}}, \quad (11)$$

where  $\text{depth}(i, j)$  is the depth of the first common father node,  $l$  is the depth of  $i$  and  $j$ , TN is the total number of word groups, and  $\text{Dis}(i, j)$  denotes the distance between  $i$  and  $j$ . For example, there are two words  $\{\text{car}, \text{wheel}\}$ , and the word group codes of  $\{\text{car}, \text{wheel}\}$  are  $\{\text{Bo21A}, \text{Bo25B}\}$ . Because two nodes are in fourth layer, the first common father node is  $\{\text{Bo}\}$ , which is in the second layer. In addition, the fourth layer contains 4223 word groups, and  $\text{Dis}(i, j)$  of  $\{\text{Bo21A}, \text{Bo25B}\}$  is 14. Therefore,  $\text{sim}(\text{Bo21A}, \text{Bo25B})$  is calculated as follows.

$$\text{sim}(\text{Bo21A}, \text{Bo25B}) = \frac{2}{4} \times \frac{4223 - 14 + 1}{4223}. \quad (12)$$

The nodes in the vocabulary network are traversed, and an edge between  $i$  and  $j$  is added when  $\text{sim}(i, j) > \text{sim}_{\min}$  (the specified threshold).

In addition, we add an edge between two nodes if an MFI in related-word set includes the words, and each MFI in related-word set is a word set with cooccurrence relations. In fact, the meaning of words in an MFI is not similar, and an MFI includes a group of words cooccurring in the same topic documents. When a text document has the words in an MFI, the text document has a high probability of belonging to certain topic. Therefore, we add an edge into the vocabulary network with low-frequency word pointing to high-frequency word.

**4.2.3. The Extraction of Feature Vectors.** In the vocabulary network, the number and the direction of edges reflect the

importance of nodes, which is similar to evaluating the importance of webpages. Thus, PageRank is utilized to obtain the importance of nodes, and the initial value  $\text{PR}_i$  of nodes is defined by

$$\text{PR}_i = \frac{f_i}{\sum_{j=1}^N f_j}, \quad (13)$$

where  $f_i$  is the frequency of word groups. After iterative computation and normalization of  $\text{PR}_i$ , we use the PageRank scores of nodes as the feature vectors of text documents instead of term frequency in this paper.

**4.3. Deep-Learning Single-Pass (DL-SP).** In this paper sparse-group DBN is proposed for dimensionality reduction of feature vectors. DBN is a model of deep learning. Luo et al. [52] found that the units of hidden layers exhibited statistical dependencies and proposed a regularization constant to restrict the relations in hidden layers. Due to the sparsity of feature vectors, we combine the word dependencies and DBN to propose a sparse-group DBN for dimensionality reduction. In addition, coverage rate (CoR) is proposed for similarity measure among feature vectors in DL-SP.

**4.3.1. Sparse-Group DBN.** Deep learning simulates the process of human thinking, and the result of deep learning is the distributed representation of an input vector. By analyzing feature vectors extracted from the vocabulary network, we find that there exists statistical dependency between entries of feature vectors, which means the entries of feature vectors will cooccur in the part of feature vectors. The word dependency is also mentioned by many researchers in previous literatures [5, 18, 53]. Cooccurrence relations are typically collected in feature vectors, which means a unique word commonly referring to “target word”, and the word dependency is quantified to measure words similarity in text clustering. We provide an example, which is the part of a feature vector in Table 3.

Because the documents in the same topic usually include related words, a part of units in visible layer is active simultaneously, and accordingly the documents in different topics usually activate different part of units. Based on this observation, we add a regularization constant to the log-likelihood of training data to retain these relations. In experiments, we use different topic documents to train the sparse-group DBN. The sequence of units in output layer is adjusted accordingly, and the cooccurring units are divided into one group. In other words, the feature vectors of different topic documents can activate different group of units in output layer. The structure of sparse-group DBN is shown in Figure 5.

Sparse-group DBN is comprised of several RBMs, and two adjacent layers are an RBMs. For retaining the dependency of the units in output layer, we define the activation probability of each group. Given a group  $S = \{y_1, y_2, \dots, y_s\}$  and training sample  $v^{(k)}$ , the group probability  $P_S(\cdot)$  is given by

$$P_S(v^{(k)}) = \sqrt{\sum_{s \in S} P(y_s = 1 | v^{(k)})^2}. \quad (14)$$

TABLE 3: The word dependencies of a feature vector.

Hj47A01	Hg19C01	Ba03A18	Ba08A07	Dm05A01	Hg01A01	Ae13B01
0.32	0.17	0.12	0.04	0	0.02	0
0.17	0.21	0.07	0.11	0	0	0
0.14	0.23	0.17	0.09	0	0	0
0.23	0.12	0.06	0.14	0	0.01	0
0	0	0	0	0.12	0.34	0.20
0	0	0	0	0.24	0.21	0.10
0	0	0	0	0.13	0.14	0.09

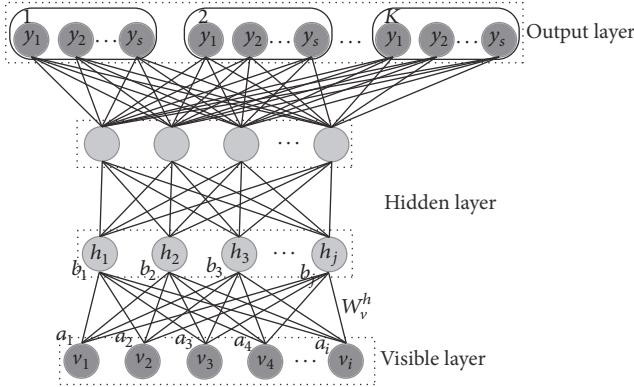


FIGURE 5: The structure of sparse-group DBN.

The output layer of the sparse-group DBN is divided into  $K$  groups, and the probability of output layer  $P_{\text{ol}}(\cdot)$  is defined by

$$P_{\text{ol}}(v^{(k)}) = \sum_{k=1}^K \sqrt{\sum_{s \in S} P(y_s = 1 | v^{(k)})^2}. \quad (15)$$

We add a regularization constant  $\lambda$  and  $P_{\text{ol}}(v^{(k)})$  to optimization function, which is maximum likelihood estimate of energy function of an RBM. The optimization function is defined by

$$\max_{W,b,c} \sum \log P(v^{(k)}) - \lambda \sum_{k=1}^K \sqrt{\sum_{s \in S} P(y_s = 1 | v^{(k)})^2}. \quad (16)$$

Equation (11) is improved to (21) accordingly, and  $\nabla W_{ij}^{(k)}$  is defined by

$$\nabla W_{ij}^{(k)} = E[v_i^{(k)} h_j^{(k)}]_{\text{data}} - E[v_i^{(k)} h_j^{(k)}]_{\text{Gibbs}} - \lambda \cdot \alpha, \quad (17)$$

where  $\alpha = (\partial/\partial(W_{ij}^{(k)}))P_S(v^{(k)}) = P(y_s = 1 | v^{(k)})/P_S(v^{(k)}) \cdot (\partial/\partial(W_{ij}^{(k)}))P(y_s = 1 | v^{(k)}) = (P(y_s = 1 | v^{(k)})^2 \cdot P(y_s = 0 | v^{(k)}) \cdot v^{(k)})/P_S(v^{(k)})$ . Accordingly, the gradient of  $(\nabla a_i^{(k)}, \nabla b_j^{(k)})$  is defined by

$$\begin{aligned} \nabla a_i^{(k)} &= E[v_i^{(k)}]_{\text{data}} - E[v_i^{(k)}]_{\text{Gibbs}} - \lambda \cdot \alpha, \\ \nabla b_j^{(k)} &= E[h_j^{(k)}]_{\text{data}} - E[h_j^{(k)}]_{\text{Gibbs}} - \lambda \cdot \alpha. \end{aligned} \quad (18)$$

**4.3.2. Similarity Measure of DL-SP.** Single-Pass is a partitional clustering algorithm. The first document is treated as the first cluster in Single-Pass, and similarity is computed between new document and existing clusters, which decides new document to join the existing cluster or to create a new cluster in terms of specified threshold. The output of sparse-group DBN is binary, and Euclidean distance and Cosine angle distance are not suitable for similarity measure in DL-SP. Therefore, we use coverage rate (CoR) for similarity measure, and CoR is defined by

$$\text{CoR}(C, d) = \frac{|C \cap d|}{C}, \quad (19)$$

where  $C = (c_1, c_2, \dots, c_n)$  is the feature vector of a cluster (named topic feature vector) and  $d = (d_1, d_2, \dots, d_n)$  is the feature vector of new document.

Moreover, the addition of many text documents to clusters has an influence on topic feature vector. In our work, we introduce optional topic feature vector  $C' = (c'_1, c'_2, \dots, c'_n)$  and the weight of feature vector to solve this problem. We provide an example of optional topic feature vector in Figure 6.

When the weight of optional topic feature vector is greater than a specified threshold in each time interval, we replace topic feature vector with optional topic feature vector as new cluster center. The weight of topic feature vector is defined by

$$w_{C'} = \frac{\sum_C f(c_i)}{\sum_C f(c_i) + \sum_{C'} f(c'_j)} - \lambda e^{k(t-t_0)}, \quad (20)$$

where  $\lambda e^{k(t-t_0)}$  is time damping function, and  $f(c_i)$  is frequency function.

## 5. Experimental Analysis

In this section, we conduct three sets of experiments to validate the effectiveness of the proposed approach, including the efficiency of FP MAX-RS in related-word set mining, the comparison of feature vectors, and the comparison of DL-SP efficiency. In this work, three Chinese text corpora, *TanCorpV1.0*, *Encyclopedia of China*, and *Sogou Corpus*, are used as the experimental datasets.

**5.1. The Efficiency of FP MAX-RS in Related-Word Set Mining.** This section is to compare running time of FP MAX and

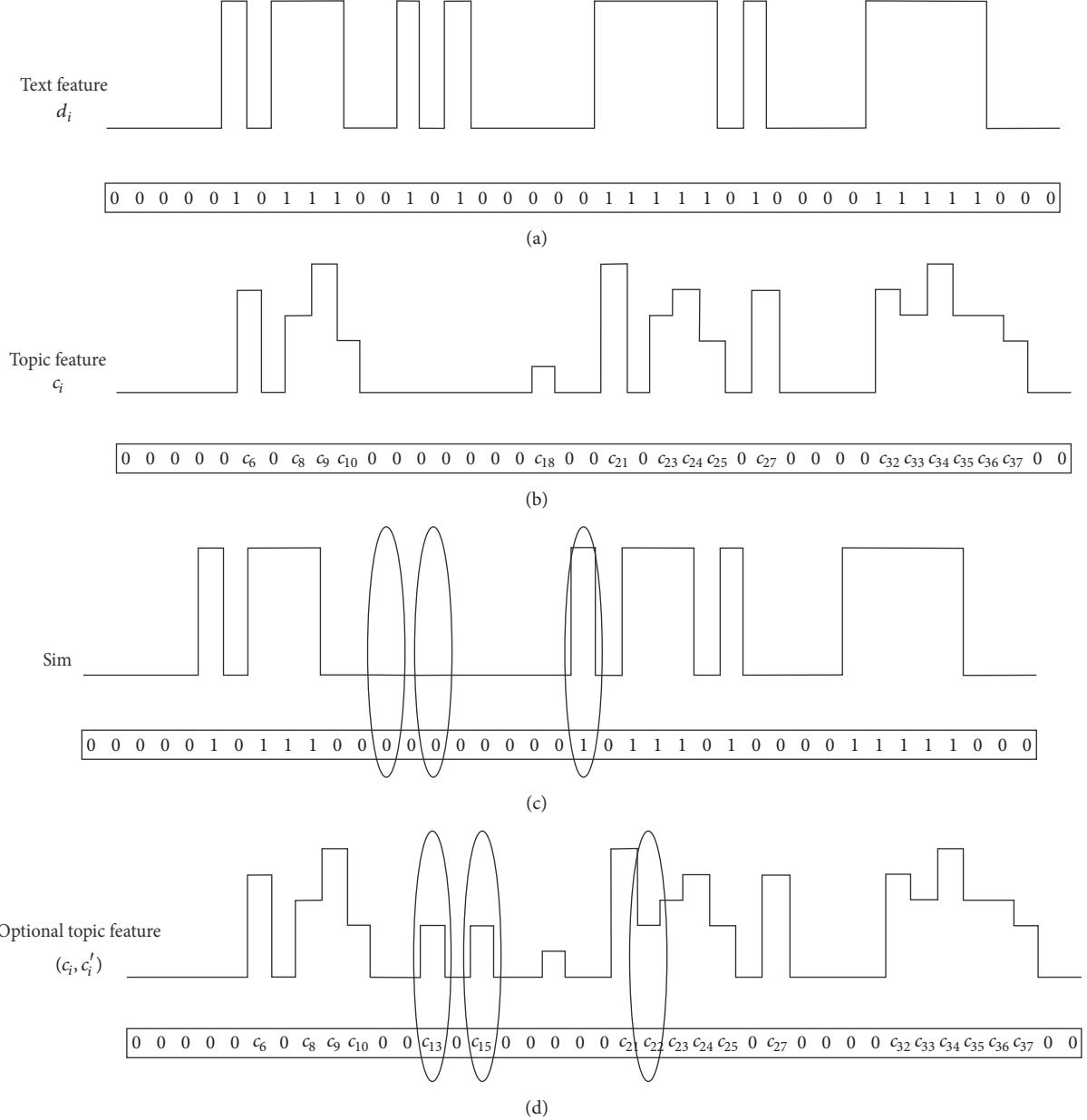


FIGURE 6: An example of optional topic feature vector.

FPMAX-RS in related-word set mining. We choose seven categories (*museum*, *property*, *education*, *military*, *car*, *sport*, and *health*) of text documents from the datasets, and each category has 50 articles. The result of experiment is listed in Figure 7.

FPMAX generates a larger amount of maximal frequent itemsets and traverses all MFI-trees for subset checking, which has an influence on the running time of FPMAX. Compared with FPMAX, FPMAX-RS has higher efficiency when  $\text{sup}_{\min}$  is smaller.

**5.2. The Comparison of Feature Vectors.** In this work, we compare the distance among the feature vectors based on

tf-idf, FC-VSM [12], and DLVN. We randomly choose two documents from the category *museum* and one document in other categories including *property*, *education*, and *military*. The aim of feature extraction is to extract the feature vectors that can represent the meaning of text documents. In other words, feature vectors in different categories have longer distance. Therefore, we compute the Euclidean distance of feature vectors in different categories based on tf-idf, FC-VSM, and DLVN. Table 4 shows the results in different categories of text documents.

In the following experiment, feature vectors are extracted based on tf-idf, FC-VSM, and DLVN. Then, *k-means* is applied for clustering. We evaluate clustering performance

TABLE 4: Euclidean distance comparison.

Category	Documents	tf-idf	Distance FC-VSM	DLVN
museum	museum <sub>1</sub> - museum <sub>2</sub>	13.02	10.49	9.17
	property - museum <sub>1</sub>	12.85	13.47	13.59
property	property - museum <sub>2</sub>	15.93	15.86	16.87
	education - museum <sub>1</sub>	14.68	14.61	14.72
education	education - museum <sub>2</sub>	11.39	11.33	12.07
	military - museum <sub>1</sub>	15.56	16.49	16.58
military	military - museum <sub>2</sub>	13.69	14.03	18.41

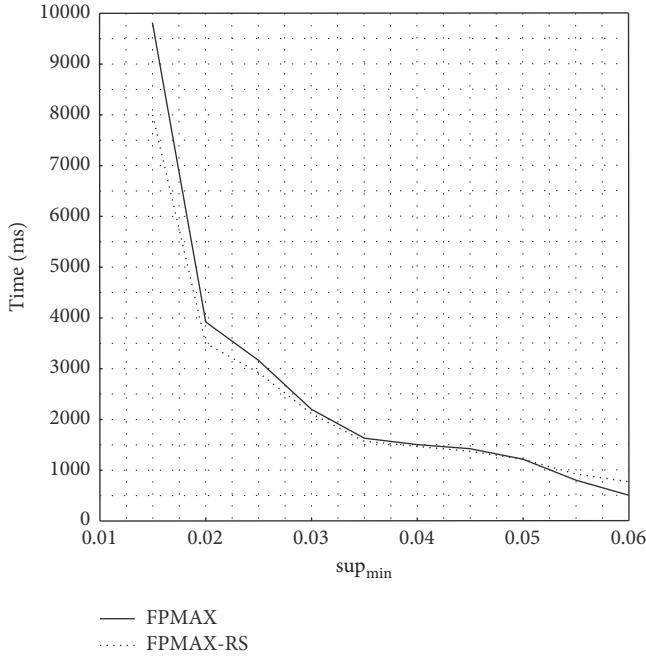


FIGURE 7: The comparison of running time.

with  $\mathcal{F}$ \_measure. Let  $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$  be clustering result and  $\mathcal{D}^* = \{d_1^*, d_2^*, \dots, d_n^*\}$  be standard dataset.  $\mathcal{F}$ \_measure is defined by

$$\mathcal{F}\text{-measure} = \frac{2 \times \mathcal{P}(\mathcal{D}, \mathcal{D}^*) \times \mathcal{R}(\mathcal{D}, \mathcal{D}^*)}{\mathcal{P}(\mathcal{D}, \mathcal{D}^*) + \mathcal{R}(\mathcal{D}, \mathcal{D}^*)}, \quad (21)$$

where  $\mathcal{P}(\mathcal{D}, \mathcal{D}^*) = |\mathcal{D} \cap \mathcal{D}^*|/|\mathcal{D}|$  is precision and  $\mathcal{R}(\mathcal{D}, \mathcal{D}^*) = |\mathcal{D} \cap \mathcal{D}^*|/|\mathcal{D}^*|$  is recall.

Because seven categories of text documents are chosen in our experiment, the specified number of clusters  $k$  is 7. Figure 8 illustrates that feature vectors based on DLVN have better performance.

**5.3. The Comparison of DL-SP Efficiency.** In this experiment, we choose text documents from the datasets, and the number of each category is listed in Table 5.

The aim of the experiment is to compare DL-SP with LSI and Single-Pass. The sparse-group DBN has 3 layers, and the

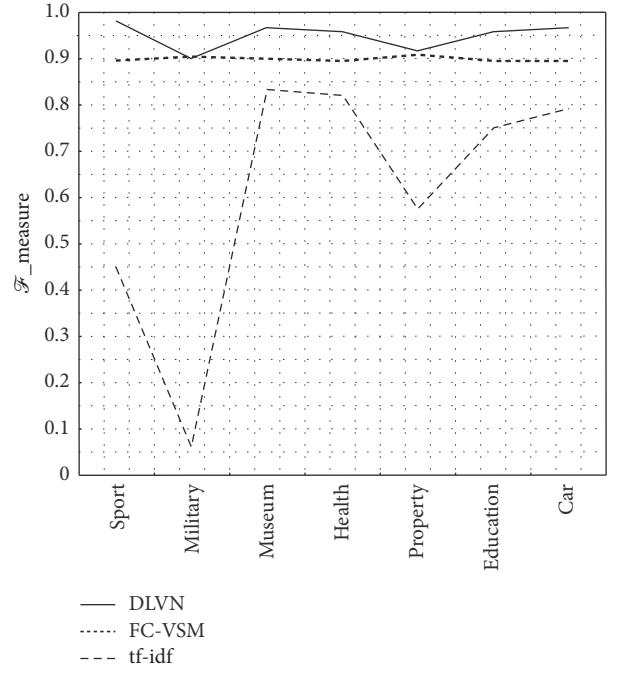
FIGURE 8: The comparison of  $\mathcal{F}$ \_measure.

TABLE 5: The datasets of experiment.

Category	The number of text documents
sport	1300
military	1500
health	1400
property	900
education	800
car	500

number of each layer is 4223, 3500, and 3000. In addition, the group number  $K$  of top layer is 200. The structure of sparse-group DBN is shown in Figure 9.

The experimental result is shown in Figure 10. DL-SP has better performance than LSI and Single-Pass in *sport*, *military*, *property*, *education*, and *health*. However,  $\mathcal{F}$ \_measure of DL-SP is lower than LSI and Single-Pass in category *car* due

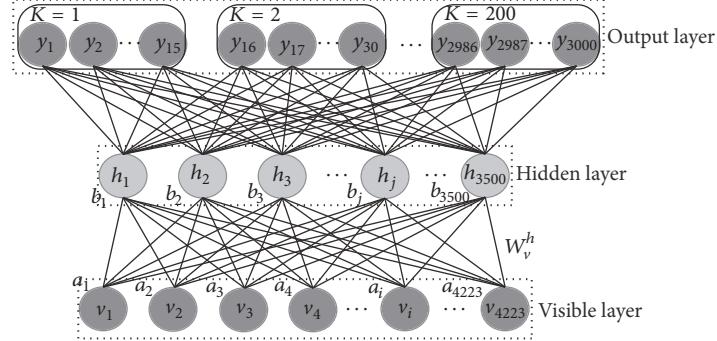


FIGURE 9: The structure of sparse-group DBN.

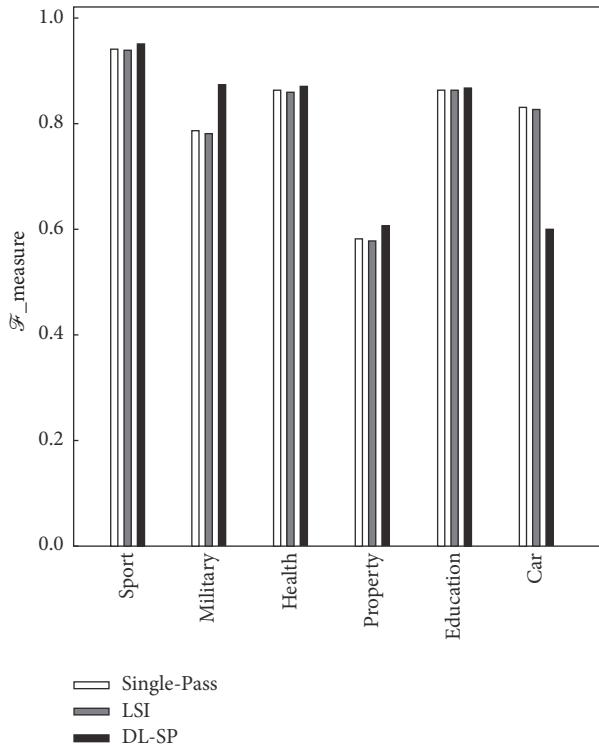


FIGURE 10: The comparison of DL-SP.

TABLE 6: The running time of DL-SP and Single-Pass.

	The dimensionality of feature vectors	Running time (s)
Single-Pass	4223	38.66
DL-SP	3000	10.84

to the smaller number of documents not training the sparse-group DBN effectively.

In this subsection, we compare the running time of DL-SP and Single-Pass, and the result is listed in Table 6.

## 6. Conclusions

In this paper, we propose an approach DLVN for text clustering. The existing term frequency-based methods only calculate the number of words, but the relations of words are not considered in feature extraction. The approach constructs vocabulary network to mine the importance of words using related-word set, which contains “cooccurrence” relations of words. Therefore, the text features of documents in the same category have shorter distance, and feature vectors have longer distance among different categories. Moreover, we employ sparse-group DBN to reduce the dimensionality of feature vectors in terms of the group relations of words. Thus, sparse-group DBN can retain the word dependency in dimensionality reduction. In the experiments, we compare the approach with well-known methods to verify our work, and the results show the performance of DLVN.

In current work, we verify the approach using Chinese corpora. We will use English text to prove the approach effectiveness in the future work. Moreover, in the process of dimension reduction, we need to train the sparse-group DBN using a large amount of text documents to improve its performance.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work has been supported by Projects U1536116 and U1636208 funded by National Natural Science Foundation of China (NSFC).

## References

- [1] A. Trabelsi and O. R. Zaiane, “Extraction and clustering of arguing expressions in contentious text,” *Data and Knowledge Engineering*, vol. 100, pp. 226–239, 2015.
- [2] K. Schouten and F. Frasincar, “Survey on aspect-level sentiment analysis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, 2016.
- [3] M. Tsytserau and T. Palpanas, “Survey on mining subjective data on the web,” *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 478–514, 2012.

- [4] S.-J. Lee and J.-Y. Jiang, "Multilabel text categorization based on fuzzy relevance clustering," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1457–1471, 2014.
- [5] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao, "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification," *Neurocomputing*, vol. 174, pp. 806–814, 2016.
- [6] W. Zhang, X. Tang, and T. Yoshida, "TESC: an approach to TExText classification using Semi-supervised Clustering," *Knowledge-Based Systems*, vol. 75, pp. 152–160, 2015.
- [7] A. B. Al-Saleh and M. E. B. Menai, "Automatic Arabic text summarization: a survey," *Artificial Intelligence Review*, vol. 45, no. 2, pp. 203–234, 2016.
- [8] F. Atefah and W. Khreich, "A survey of techniques for event detection in Twitter," *Computational Intelligence*, vol. 31, no. 1, pp. 132–164, 2015.
- [9] G. Stilo and P. Velardi, "Efficient temporal mining of micro-blog texts and its application to event discovery," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 372–402, 2016.
- [10] G. Huang, J. He, Y. Zhang et al., "Mining streams of short text for analysis of world-wide event evolutions," *World Wide Web*, vol. 18, no. 5, pp. 1201–1217, 2014.
- [11] U. Erra, S. Senatore, F. Minnella, and G. Caggianese, "Approximate TF-IDF based on topic extraction from massive message stream using the GPU," *Information Sciences*, vol. 292, pp. 143–161, 2015.
- [12] C. Qimin, G. Qiao, W. Yongliang, and W. Xianghua, "Text clustering using VSM with feature clusters," *Neural Computing and Applications*, vol. 26, no. 4, pp. 995–1003, 2015.
- [13] J. Martinez-Gil, "An overview of textual semantic similarity measures based on web intelligence," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 935–943, 2012.
- [14] K. K. Bharti and P. K. Singh, "Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3105–3114, 2015.
- [15] L. Yue, W. Zuo, T. Peng, Y. Wang, and X. Han, "A fuzzy document clustering approach based on domain-specified ontology," *Data and Knowledge Engineering*, vol. 100, pp. 148–166, 2015.
- [16] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao, "A semantic approach for text clustering using WordNet and lexical chains," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2264–2275, 2015.
- [17] L. Bing, S. Jiang, W. Lam, Y. Zhang, and S. Jameel, "Adaptive concept resolution for document representation and its applications in text mining," *Knowledge-Based Systems*, vol. 74, no. 1, pp. 1–13, 2015.
- [18] R. Irfan, C. K. King, D. Grages et al., "A survey on text mining in social networks," *Knowledge Engineering Review*, vol. 30, no. 2, pp. 157–170, 2015.
- [19] N. Indurkhy, "Emerging directions in predictive text mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 4, pp. 155–164, 2015.
- [20] M. T. Mills and N. G. Bourbakis, "Graph-based methods for natural language processing and understanding—a survey and analysis," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 44, no. 1, pp. 59–71, 2014.
- [21] H. Mousavi, D. Kerr, M. Iseli, and C. Zaniolo, "Mining semantic structures from syntactic structures in free text documents," in *Proceedings of the 8th IEEE International Conference on Semantic Computing (ICSC '14)*, pp. 84–91, IEEE, Newport Beach, Calif, USA, June 2014.
- [22] S. Jun, S.-S. Park, and D.-S. Jang, "Document clustering method using dimension reduction and support vector clustering to overcome sparseness," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3204–3212, 2014.
- [23] W. Z. Zhu and R. B. Allen, "Document clustering using the LSI subspace signature model," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 4, pp. 844–860, 2013.
- [24] X. Wu, X. Chen, X. Li, L. Zhou, and J. Lai, "Adaptive subspace learning: an iterative approach for document clustering," *Neural Computing & Applications*, vol. 25, no. 2, pp. 333–342, 2014.
- [25] H. Kriegel and E. Ntoutsi, "Clustering high dimensional data," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 2, pp. 1–8, 2014.
- [26] K. K. Bharti and P. K. Singh, "Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering," *Applied Soft Computing Journal*, vol. 43, pp. 20–34, 2016.
- [27] M. C. N. Barioni, H. Razente, A. M. R. Marcelino, A. J. M. Traina, and C. Traina, "Open issues for partitioning clustering methods: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 3, pp. 161–177, 2014.
- [28] J.-P. Mei and L. Chen, "Proximity-based k-partitions clustering with ranking for document categorization and analysis," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7095–7105, 2014.
- [29] V. Tunali, T. Bilgin, and A. Camurcu, "An improved clustering algorithm for text mining: multi-cluster spherical K-means," *International Arab Journal of Information Technology*, vol. 13, no. 1, pp. 12–19, 2016.
- [30] Y. Li, C. Luo, and S. M. Chung, "A parallel text document clustering algorithm based on neighbors," *Cluster Computing*, vol. 18, no. 2, pp. 933–948, 2015.
- [31] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.
- [32] T. Peng and L. Liu, "A novel incremental conceptual hierarchical text clustering method using CFu-tree," *Applied Soft Computing*, vol. 27, pp. 269–278, 2015.
- [33] Q. Chen, J. F. Lu, and H. Zhang, "A text mining model based on improved density clustering algorithm," in *Proceedings of the 4th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC '13)*, Beijing, China, November 2013.
- [34] S. F. Hussain, M. Mushtaq, and Z. Halim, "Multi-view document clustering via ensemble method," *Journal of Intelligent Information Systems*, vol. 43, no. 1, pp. 81–99, 2014.
- [35] A. Wahid, X. Gao, and P. Andreae, "Multi-view clustering of web documents using multi-objective genetic algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '14)*, pp. 2625–2632, Beijing, China, July 2014.
- [36] X. Pei, T. Wu, and C. Chen, "Automated graph regularized projective nonnegative matrix factorization for document clustering," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1821–1831, 2014.
- [37] M. Lu, X.-J. Zhao, L. Zhang, and F.-Z. Li, "Semi-supervised concept factorization for document clustering," *Information Sciences*, vol. 331, pp. 86–98, 2016.
- [38] C.-K. Yau, A. Porter, N. Newman, and A. Suominen, "Clustering scientific documents with topic modeling," *Scientometrics*, vol. 100, no. 3, pp. 767–786, 2014.

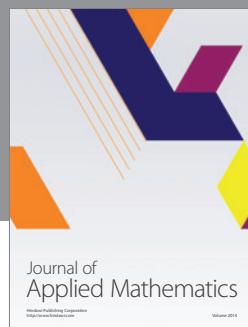
- [39] Y. Ma, Y. Wang, and B. Jin, “A three-phase approach to document clustering based on topic significance degree,” *Expert Systems with Applications*, vol. 41, no. 18, pp. 8203–8210, 2014.
- [40] C. C. Aggarwal, Y. Zhao, and P. S. Yu, “On the use of side information for mining text data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 6, pp. 1415–1429, 2014.
- [41] R. M. Marcacini, M. A. Domingues, E. R. Hruschka, and S. O. Rezende, “Privileged information for hierarchical document clustering: a metric learning approach,” in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR '14)*, pp. 3636–3641, August 2014.
- [42] L. Cagnina, M. Errecalde, D. Ingaramo, and P. Rosso, “An efficient particle swarm optimization approach to cluster short texts,” *Information Sciences*, vol. 265, pp. 36–49, 2014.
- [43] W. Song, Y. Qiao, S. C. Park, and X. Qian, “A hybrid evolutionary computation approach with its application for optimizing text document clustering,” *Expert Systems with Applications*, vol. 42, no. 5, pp. 2517–2524, 2015.
- [44] R. Forsati, A. Keikha, and M. Shamsfard, “An improved bee colony optimization algorithm with an application to document clustering,” *Neurocomputing*, vol. 159, no. 1, pp. 9–26, 2015.
- [45] K. K. Bharti and P. K. Singh, “Chaotic gradient artificial bee colony for text clustering,” *Soft Computing*, vol. 20, no. 3, pp. 1113–1126, 2016.
- [46] F. Wang and J. Sun, “Survey on distance metric learning and dimensionality reduction in data mining,” *Data Mining and Knowledge Discovery*, vol. 29, no. 2, pp. 534–564, 2014.
- [47] Y.-S. Lin, J.-Y. Jiang, and S.-J. Lee, “A similarity measure for text classification and clustering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1575–1590, 2014.
- [48] M. T. Hassan, A. Karim, J.-B. Kim, and M. Jeon, “CDIM: document clustering by discrimination information maximization,” *Information Sciences*, vol. 316, pp. 87–106, 2015.
- [49] M. T. Hassan and A. Karim, “Clustering and understanding documents via discrimination information maximization,” in *Proceedings of the Pacific-Asia Conference on Advances in Knowledge Discovery & Data Mining (PAKDD '12)*, Kuala Lumpur, Malaysia, May 2012.
- [50] D. Cai and C. J. van Rijsbergen, “Learning semantic relatedness from term discrimination information,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 1860–1875, 2009.
- [51] G. Grahne and J. Zhu, “High performance mining of maximal frequent itemsets,” in *Proceedings of the SIAM Workshop High Performance Data Mining: Pervasive and Data Stream Mining (HPDM:PDS '03)*, San Francisco, Calif, USA, May 2003.
- [52] H. Luo, R. Shen, and C. Niu, “Sparse group restricted boltzmann machines,” in *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI '11)*, San Francisco, Calif, USA, August 2011.
- [53] S. Padó and M. Lapata, “Dependency-based construction of semantic space models,” *Computational Linguistics*, vol. 33, no. 2, pp. 161–199, 2007.



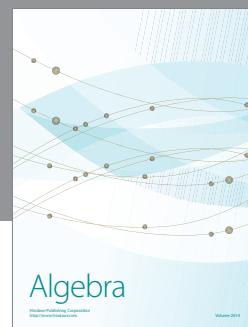
Advances in  
Operations Research



Advances in  
Decision Sciences



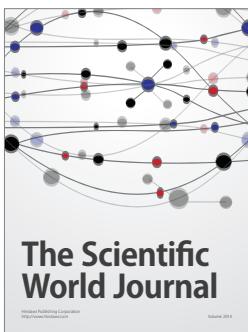
Journal of  
Applied Mathematics



Algebra



Journal of  
Probability and Statistics



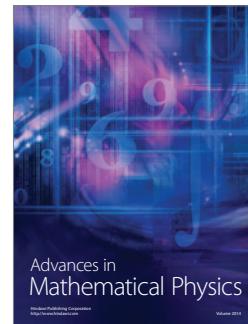
The Scientific  
World Journal



International Journal of  
Differential Equations



International Journal of  
Combinatorics



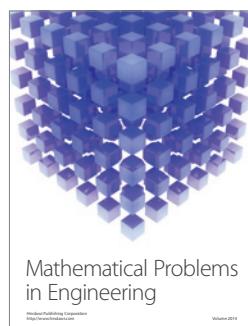
Advances in  
Mathematical Physics



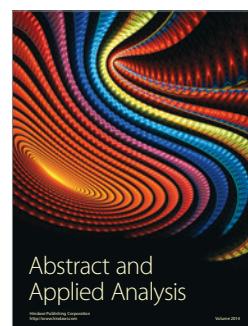
Journal of  
Complex Analysis



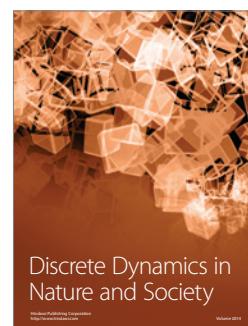
Journal of  
Mathematics



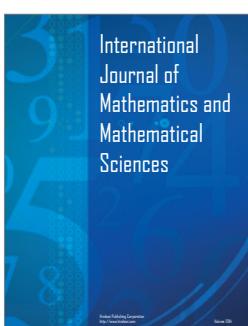
Mathematical Problems  
in Engineering



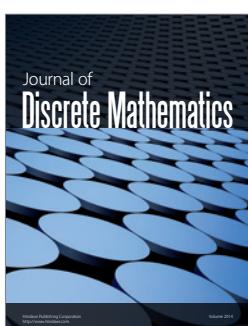
Abstract and  
Applied Analysis



Discrete Dynamics in  
Nature and Society



International  
Journal of  
Mathematics and  
Mathematical  
Sciences



Journal of  
Discrete Mathematics



Journal of  
Function Spaces



International Journal of  
Stochastic Analysis



Journal of  
Optimization