IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 11, 2020, accepted April 23, 2020, date of publication April 28, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991074

Network-Based Bag-of-Words Model for Text Classification

DONGYANG YAN, KEPING LI, SHUANG GU, AND LIU YANG
State Key Laboratory of Rail Traffic Control and Safety, Beijing 100044, China

Corresponding author: Keping Li (kpli@bjtu.edu.cn)

This work was supported in part by the Beijing Natural Science Foundation under Grant 8202039, and in part by the National Key Research and Development Program of China under Grant 2017YFB1201105.

ABSTRACT The rapidly developing internet and other media have produced a tremendous amount of text data, making it a challenging and valuable task to find a more effective way to analyze text data by machine. Text representation is the first step for a machine to understand the text, and the commonly used text representation method is the Bag-of-Words (BoW) model. To form the vector representation of a document, the BoW model separately matches and counts each element in the document, neglecting much correlation information among words. In this paper, we propose a network-based bag-of-words model, which collects high-level structural and semantic meaning of the words. Because the structural and semantic information of a network reflects the relationship between nodes, the proposed model can distinguish the relation of words. We apply the proposed model to text classification and compare the performance of the proposed model with different text representation methods on four document datasets. The results show that the proposed method achieves the best performance with high efficiency. Using the Eccentricity property of the network as features can get the highest accuracy. We also investigate the influence of different network structures in the proposed method. Experimental results reveal that, for text classification, the dynamic network is more suitable than the static network and the hybrid network.

INDEX TERMS Bag-of-words, classification, complex network, text correlation, KNN.

I. INTRODUCTION

During the last decades, people have witnessed the impact of the advancement of information technology. The rapid development of social media on the internet has been producing more and more information, in which text information plays a significant role. Meanwhile, a typical scenario is how to classify text data into topic sets by computer so that people can conveniently search the data they want. The text classification task, which assigns the documents to the best-suited topic, has drawn much attention from researchers.

A typical text classification work includes text preprocessing, feature selection, feature extraction, similarity computation, and classifier determination [1]. Though owing to the advantage in understanding human language, it is natural for people to judge whether a document belongs to a particular topic directly by reading and understanding, this process is not practical for a computer. So the text classification of a computer starts with the text representation, which transfers text data to the form that is convenient for computer processing. The commonly used text representation method is the bag-of-words (BoW) model [2]–[4]. This model maps a document into a vector as $v = [x_1, x_2, \ldots, x_n]$, where x_i denotes the occurrence of the ith word in basic terms. The basic terms are collected from the datasets, which are usually the top n highest-frequency words. The value of the occurrence feature can be a binary, term frequency, or TF-IDF. A binary value denotes whether the ith word is presented in a document, which reckons without the weight of words. The term frequency is the number of occurrences of each word. Generally, the word with high frequency in a document contains the representative idea about this document, with the exception that some words may have high frequency among all documents. TF-IDF (term frequency-inverse document frequency) balances the weight of the words that always have a high frequency. It assumes that the importance of a word increases proportionally to its frequency in a document but is offset by its frequency in the whole corpus [5], [6].

Though the BoW model is a useful and straightforward method for text representation, there are still some problems.

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

The value of x_i, whether in binary, term frequency, or TF-IDF form, is matched and counted without considering the influence of others words. So the processing of text data may lose much context information without dealing with correlated words. To illustrate this limitation, we provide two simple sentences as a toy example: **Sen 1**, "a cat ate a small white mouse;" **Sen 2**, "a small white mouse ate a cat." The basic terms are (cat, eat, small, white, mouse), and for both two sentences, each word in basic terms occurs once. The BoW model will project **Sen 1** and **Sen 2** to the same vector, i.e. $v_1 = v_2 = [1, 1, 1, 1, 1]$, though the two sentences have the opposite meaning.

In this paper, we adopt the network model to overcome the limitation of the BoW model mentioned above. The complex network is now attracting much attention in the study of real-world systems [7] (such as social systems, biological systems, and authors systems). The advantage of the network model to analyze text data is that through the network tools, one can have an insight view of several features of texts, e.g., complexity [8], and symmetry [9]. By using the network model, we can take more context information of the text into account. To extend the application of the network model to BoW, we come up with a network-based strategy: Attribute of Network Extended to BoW (AEBoW). AEBoW maps documents to vectors in which the value of the corresponding word is replaced by the weight of the network node attribute. The main difference between AEBoW and BoW is that the value of x_i will not only match the frequency of the ith basic term but also match the role it plays in high-level features of the text, e.g., the structural and semantic difference. By using the Degree of the network model, the AEBoW model will project **Sen 1** and **Sen 2** to $v_1 = [1, 2, 2, 2, 1]$ and $v_2 = [1, 2, 1, 2, 2]$, which can capture the meaning difference of two sentences (see details in section IV.F).

We summarize the main contributions of this paper as follows

- ✧ We propose the AEBoW model to maintain correlated information among the words in the text.
- ✧ We demonstrate the efficiency of the AEBoW model by applying it to text classification. We also verify the performance of the proposed model by comparing it with seven text representation methods and the word embedding model (deep learning method) on four different datasets.
- ✧ We present the results of the AEBoW model based on three kinds of network tools: the dynamic network, the static network, and the hybrid network.
- ✧ By comparing the performance of the AEBoW model based on different kinds of networks, we observe the dynamic network is more suitable for the AEBoW model.

This paper is organized as follows. In Section II, we introduce some related works, including the studies on text representation and text complex networks. The proposed AEBoW model is presented in Section III. In Section IV, we give the experimental results on the performance of the proposed model and the comparison with different representation methods. We extend the proposed model to more possible applications in Section V. And, at last, we provide the concluding remarks in Section VI.

II. RELATED WORK

Because our work aims to incorporate the network model into BoW, in this section, we give a brief review of these two associated works, respectively.

A. TEXT REPRESENTATION METHODS

In the field of text data mining, text representation is the keystone for the computer to understand. Though the BoW model is simple and commonly used, it suffers from the sparsity with high dimensionality and the loss of relations among words. To improve the BoW model, researchers have proposed some methods like latent semantic analysis (LSA) [10] and topic model [11]. LSA applies the singular value decomposition (SVD) to transfer the original BoW representation to the vectors with a lower dimension. If the origin vectors are frequency-based, the transferred vectors are also approximately linearly related to the term frequency. The topic model, attaching the probability distribution of words to the topic probability distribution, though has a more mature mathematical foundation than LSA, it is still a frequency-based method, which may not be able to capture the genuine semantic relations. Being different from the BoW model, word embedding maps the words into dense and low-dimensional vectors through machine learning methods [12]–[14], e.g., multilayers neural networks. This kind of method can capture the relations of words like "king + woman $\approx$ queen." Nevertheless, the mapped vectors are learned from a large corpus of text data, making this training process very time-consuming and highly dependent on the quality of training corpus. There is also the representation model that combines word embedding model and deep learning with BoW [15], which uses the pre-trained word embeddings to get the fuzzy matching for the BoW model. The matching process is based on the whole basic terms, which is sometimes redundant (we will explain it in section **IV**). In this paper, the proposed AEBoW model is a combined method, which adopts the simplicity of the BoW model while considering the inner-correlation of words by a network tool.

B. THE NETWORK MODEL FOR TEXT ANALYSIS

In recent years, more and more works studies on the network model in analyzing human language. The network is constructed from a series of nodes connected by their interrelations. The network model has been used for different complex systems because of its simplicity and generality. Without loss of generality, the networks of text share the same properties that unveiled from other complex systems like the small-world structure and scale-free phenomena [16]–[19]. Moreover, the network properties have been proved to be a powerful tool to capture the features of texts. The out degrees, clustering coefficient, and deviation of network growth are

related to the text quality [20] while the community structures and weighted edges of the network can be used to detect the key segments [21], [22]. The topological properties of networks will help enhance the performance of several tasks (authors recognization [9], [23], text similarity [24], text summarization [25], text classification [26], and shorts text analysis [27], [28]). In recent years, the image analysis approach based on the network model is proposed to be supplementary on semantic-based applications, as the mesoscopic structure can reveal the visual ''calligraphy'' of a document [29]. The network model, when applied to text analysis, can capture subtle interactions among words, which will provide richer information than the occurrence feature.

III. THE PROPOSED MODEL

In this section, the AEBoW model is presented. It should be noted that the underlying assumption is that the node properties of the complex network can reflect their specific relevancy among other nodes. Of particular influence on the structure of the network, the linguistic units, and their relations to form edges determine the topological configuration, which affects the corresponding relevancy of nodes [18]. We introduce three different sub-structures of text complex networks: the static semantic network, the co-occurrence network, and the hybrid network. Moreover, based on the same text representation model, we compare the performance of these sub-structures in practical use in section IV.

Before going into details about the proposed model, the general steps to deal with specific problems using this model are summarized as follows:

STEP 1: Lemmatize all the words in training data, and eliminate the stop-words. Lemmatization makes the words transferred into their original forms, e.g., the nouns are converted to the singular forms, and the verbs are converted to the infinitive forms. The stop-words are words that occur high frequency with little useful semantic content.

STEP 2: For each text sample in training data and test data, represent the text as networks (the type of network is a hyper-parameter). Then get the value of particular network property at all nodes. Each node in a network is bounded to a word in correspond text sample.

STEP 3: Represent each sample as a column vector, in which the value of each element is the network property of the corresponding node that obtained in step 2. The value of the node that not included in a text sample will be replaced by '0' in the corresponding column vector. Note that the full words bag of big training data is considerably large, which causes the column vector high-dimensionality and sparse. One optional solution is to adopt the most used words in the datasets, which called basic terms, to reduce dimensionality.

STEP 4: Train the classifier using the vectors of training data obtained from step 3 as inputs.

The above steps are presented as a flowchart in figure 3.

The following part of this section will go into detail about the proposed model.

A. REPRESENTING TEXTS AS NETWORKS

Generally, the network model can be described as a graph with graph theory [16]. An undirected network that we adopt to represent text is generally represented as $G = (N, E)$, where $N = \{n_1, n_2, \ldots, n_l\}$ denotes the set of nodes (or vertices) and $E = \{e_1, e_2, \ldots, e_k\}$ denotes the set of links between particular double nodes. We can use an adjacency matrix $A = (a_{ij})_{l\times l}$ to represent graph G, in which the element a_{ij} is defined as follows:

$$a_{ij} = \begin{cases} 1 & if\ (n_i, n_j) \in E \\ 0 & if\ (n_i, n_j) \notin E \end{cases} \tag{1}$$

The appropriately represented texts as networks are the inventories of text units with organized relations among them. For example, when the text units are words, the relations among them may be the semantic relations or their positional relations in actual language use [18]. Different organized relations may lead to different network structures in terms of the same text. If the text network is modeled with the words as nodes and the words' semantic relations as edges, this kind of network, called static linguistic networks, contains relative fixed nodes relationships. Another kind of network, named dynamic linguistic network, is modeled with the links being the naturally-occurring of words in texts, reflecting much information on actual language style.

This paper introduces the co-occurrence network as the sub-network of dynamic linguistic networks, the static semantic network as the sub-network of the static linguistic network. The co-occurrence network describes the texts as the network in which the nodes (words) are joined when they co-occur within a distance [17]. Moreover, the static semantic network, describing the texts as the inventories of semantic relations, is constructed following the rule that two nodes (words) are connected when they are organized in the same class of a dictionary [18]—in this paper, this relationship is captured through the WordNet [30]. Based on the WordNet, the words as nodes are linked when they are in the same word set with hypernymy, meronymy (including the entailment of verbs), or synonymy relationship. For a combination of the above two kinds of networks, we propose the hybrid network that contains relations both in static semantic network and co-occurrence network. The hybrid network has the information held in both the dynamic network and the static network, making it more helpful in text classification work.

The process of text network construction starts with a text preprocessing. Firstly, lemmatize the words [8] (e.g., the nouns are converted to the singular forms, and the verbs are converted to the infinitive forms). Then, eliminate words with little useful semantic content, which are named as stop-words, because in some text processing like classification, these words are helpless, sometimes misleading [24]. Figure 1 shows three text networks of the following documents. A more detailed process to construct these network models is described in the Supplementary Information.

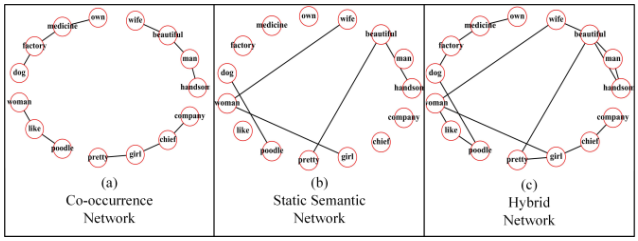

FIGURE 1. Three networks modeled by various unit relationships. (a) Co-occurrence network; (b) Static semantic network; (c) Hybrid network.

(1) This handsome man has a beautiful wife.
(2) He owns a medicine factory and a dog.
(3) This beautiful woman likes her poodle.
(4) The pretty girl is the chief of this company.

Figure 1(a) is a co-occurrence network, and figure 1(b) is a static semantic network. In figure 1(b), "handsome" and "beautiful" are synonyms; "dog" has a hypernymy relationship with "poodle." As a mixed form of both types of networks, Figure 1(c) shows a hybrid network with static and dynamic relations, which in some extents, contains complementary information.

B. TO AVOID ISOLATED NODES IN THE STATIC SEMANTIC NETWORK

The above mentioned static network of text is an ideal model for the static property: from the view of the formation process, the edges of the words have already been pre-defined in the corpus (WordNet). However, in some short texts, this kind of static network contains many isolated nodes, e.g., "factory," "medicine," and "own" in figure 1. Not only are these isolated nodes not helpful in text analysis, but they cause computing problems in a network model, e.g., the calculating of some properties of the network model requires that the network is connected. To deal with this problem, we make the following assumptions:

1. The static semantic network is not allowed to contain isolated nodes.

2. If the semantic relevancy in the WordNet is not enough to avoid the existence of isolated nodes, the nodes with no edges are randomly connected to be a circle, i.e., the isolated nodes form a sub-network with every node having two neighbors.

3. The isolated nodes are connected to the other nodes following the laws that the nodes are more likely to link to the nodes with more neighbors.

With the above assumptions, we construct the static semantic network used in this paper, as shown in figure 2. Note that assumptions 2 and 3 do not have a complete theory explanation but are only made to avoid isolated nodes without losing the unique information of other nodes. Assumption 2 guarantees that the isolated nodes are homogeneous (the nodes in the sub-network of isolated nodes all contain two neighbors). Assumption 3 retains the disassortativity of text networks [18], which means the weakly linked nodes are more likely to attach to nodes with a large degree. The nodes connected with edges formed in semantic relevancy are the same as figure 1(b) shows, and the other nodes which have no neighbors are connected to the network with the laws described in assumption 2 and 3.

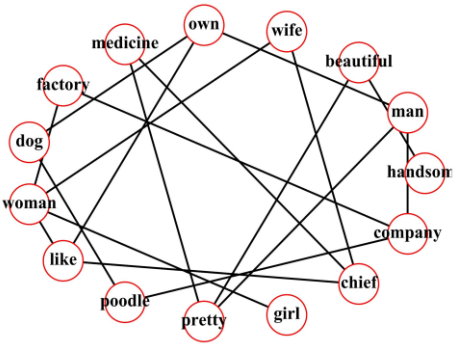

FIGURE 2. Example of the static semantic network with laws to avoid isolated nodes.

C. AEBoW: A REPRESENTATION OF THE INTER-CORRELATION AMONG WORDS

The AEBoW (Attribute of Network Extended to BoW) model is a simple extension of the BoW model, where the mapped vectors contain the elements with the value being a particular attribute of the network. The attributes of the network, which are also named the properties, are the fundamental quantities used to describe the structure properties (or topology) of a network.

For a document (denote as d with the corresponding network model g_d), the representation by the AEBoW is $z^d = [z_1^d, z_2^d, \ldots, z_n^d]$, where z_i^d is defined as

$$z_i^d = \begin{cases} f_{g_d}^a(w_i), & \text{if } w_i \text{in} d. \\ 0, & \text{else.} \end{cases} \tag{2}$$

In (2), f_g^d is the function that returns the value of an individual node against the property a and network model g_d; w_i denotes the ith word in the basic terms.

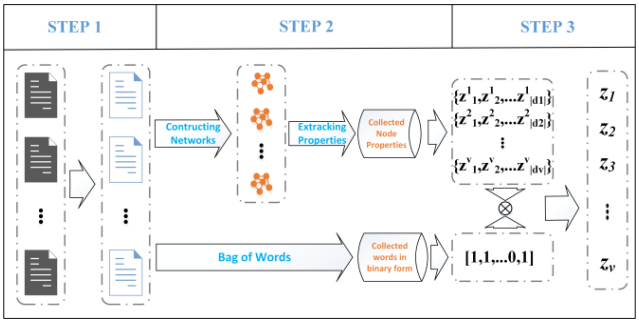

FIGURE 3. AEBoW framework steps.

We show the process of the AEBoW model in figure 3. Firstly, the documents are transformed into networks, and

the kind of networks (static, dynamic, or hybrid) should be pre-defined. The idea of the BoW model is used to collect the words among all the documents in binary form. Then the extracted properties are located to the corresponding place. We also list the procedure of AEBoW in Algorithm 1. An illustration of AEBoW by a toy example is shown in figure 4. The pseudo samples – d_1 "A cat is sitting on the table while a dog is running towards it" and d_2 "A cat and a dog were both sitting on the table, and the dog ran away later" – are represented as vectors of AEBoW model. The vector mapped from d_1 is [1, 2, 2, 2, 1, 0, 0] because the Degree of node 'cat,' 'dog,' 'sit,' 'table' and 'run' is 1, 2, 2, 2, 1, respectively, while 'away' and 'later' do not occur in d_1. Similarly, the vector of d_2 is projected.

Algorithm 1 AEBoW Framework

Inputs: Text corpus $\boldsymbol{T}$ including v documents, network property $\boldsymbol{a}$, and the network type $\boldsymbol{g}$.
Outputs: Text vectors $\boldsymbol{Z}$ of $\boldsymbol{T}$.

1. Collect the basic terms $\boldsymbol{B}$ based on the frequency that words occur in $\boldsymbol{T}$.
2. **for** d in $\boldsymbol{T}$:
   Construct the network $\boldsymbol{g_d}$ of d:
   for *node* in $\boldsymbol{g_d}$:
   Get the index i of *node* in $\boldsymbol{Z}$
   if *node* in $\boldsymbol{B}$:
   $Z_i^d = f_{g_d}^a(node)$
   else:
   $Z_i^d = 0$
   end if
   end for
   end for
3. return $\boldsymbol{Z}$

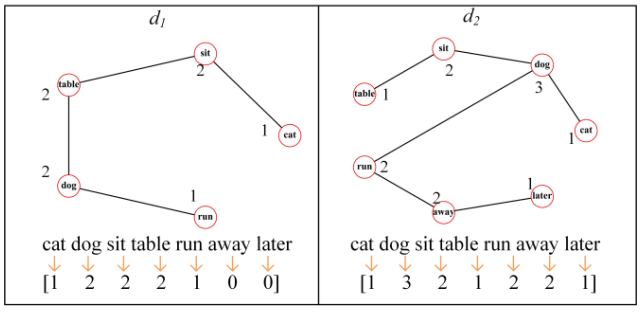

FIGURE 4. A toy example about text representing based on the AEBoW model (for Degree property) through the co-occurrence network.

The development of complex networks has induced various indexes for the observed properties of real networks, e.g., node degree, betweenness, and clustering [16]. Though there are various property measures, the experimental results show that not all of them are suitable for text classification. The following part of this sub-section introduces network properties that perform well in the experimental results.

Degree: The degree k_i of a node i is the number of its neighbor nodes or the edges incident with it in the complex network. The Degree denotes the connectivity of a node, which shows the ability to integrate with other nodes. For an undirected graph, given the adjacency matrix A, the degree k_i of node i is defined as

$$k_i = \sum_{j=1}^{N} a_{ij}, \tag{3}$$

where N is the size of matrix A, i.e., the number of nodes in the complex network. In the matrix A, the element a_{ij} is binary value denoting that whether node i and node j is connected through an edge.

Eccentricity: The eccentricity ec_i of a node i is the maximum distance from e_i to other nodes in the complex network. For a network G, the eccentricity ec_i is defined as

$$ec_i = \max_{j \in N \setminus n_i} l_{ij}, \tag{4}$$

where l_{ij} is the shortest distance from node i to node j. In some cases, the text network may be disconnected, which means that the network contains more than one part without links between each other. In this paper, for convenience, we assume that the eccentricity ec_i of the network that is not connected is the maximum distance from e_i to its reachable nodes.

PageRank: PageRank is initially designed for ranking web pages based on the directed graph [31]. The idea is that the more web pages that a page is pointed to and the more critical the pointing webs are, the more weighted this pointed page is. The definition is a voting process, which needs recursive computing. The rank of a given node (page) i is defined to be

$$r(i) = \sum_{j \in P_i} \frac{r(j)}{num(j)}, \tag{5}$$

where P_i is the set of nodes that point to i, and $num(j)$ is the number of links that point out from j in graph G. For a start, we can arbitrarily assign the ranking to all the nodes of graph G, e.g., $r_0(i) = 1/l$, $i \in N$, and successively update the ranks of the nodes by (5). In this paper, we adopt this method to the undirected graph by assuming that each undirected edge (i, j) is equal to two directed edges $i \to j$ and $j \to i$.

Accessibility: This concept is used to measure the ability of a node to reach the number of nodes after h steps implemented through self-avoiding random walks [38]. It is mathematically defined as

$$\alpha_h(i) = \exp\left(-\sum_j P^{(h)}(i,j) \log P^{(h)}(i,j)\right), \tag{6}$$

where $P^{(h)}(i, j)$ denotes the possibility of node i reach node j after h steps. The accessibility measures the influence of a node in the complex network, i.e., the nodes playing more critical roles usually can access more neighbors.

IV. EXPERIMENTAL RESULTS

In this section, we apply the AEBoW model in text classification. The proposed method is compared with seven

text representation methods on four datasets. Furthermore, we also compare AEBoW with the deep learning algorithm at the end of this section.

A. DATASETS DESCRIPTION

There are four datasets used in the experiments.

20Newgroups is a group of news with nearly 20000 documents and 20 news topics. This dataset is kindly preprocessed in [32], [33].

WebKB is collected from webpages by the World Wide Knowledge Base project [32]. The training data and testing data of these documents were predesignated in [33], [34]: 2803 documents for training and 1396 documents for testing.

Reuters 52 is extracted from *Reuters 21578* by [32]. This dataset includes 52 categories, deleting some categories of *Reuters 21578* that contain only a few documents.

Amazon Reviews contains 10000 labeled reviews with 2 categories. The original dataset can be found in [35].

We list the details of these datasets in table 1. Note that all the datasets are preprocessed by removing the stop words and lemmatizing.

TABLE 1. The description of four datasets.

Datasets	20NG	WebKB	R52	Amazon
Topics	20	4	52	2
Train docs	11293	2803	6532	6666
Test docs	7528	1396	2568	3334
Total	18821	4199	9100	10000

Notes: 20NG-20Newgroups; R52-Reuters 52; Amazon-Amazon Reviews

B. EXPERIMENTAL SETUP

The classification work is done by KNN measure [36], and the similarity distance is computed through the cosine similarity [24]. Classification accuracy [37] is used to evaluate performance. Firstly, we briefly describe the KNN measure, cosine similarity, and classification accuracy.

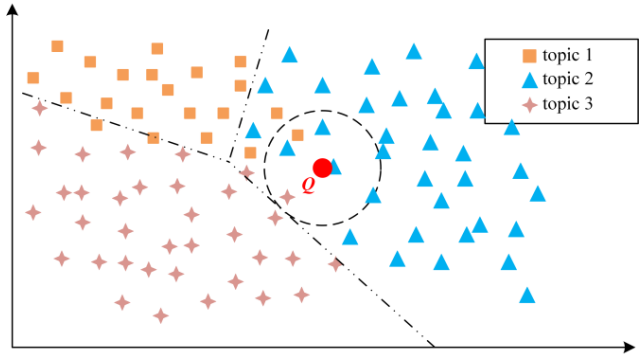

FIGURE 5. An illustration of KNN. For k = 5, most of the nearest neighbors of the circle node Q belong to topic 2. So node Q is more likely to belong to topic 2.

KNN: The KNN (k-Nearest-Neighbors) is a simple and effective non-parametric classification method. The idea of KNN, as shown in figure 5, is that a node in space is more likely to be the same type as the nodes occur most in its k nearest neighbors, which are captured based on particular similarity distance. Because this method is parameter-free except k, making it a lazy learning method, it is used in many applications.

Cosine similarity: The cosine similarity computes the similarity distance of two vectors in space. For vector $v_i = [v_{i1}, v_{i2}, \ldots, v_{il}]$ and $v_j = [v_{j1}, v_{j2}, \ldots, v_{jl}]$, the cosine similarity is defined as

$$c_{ij} = \frac{v_i \cdot v_j}{|v_i||v_j|} = \frac{\sum_{k=1}^{l} v_{ik} v_{jk}}{\sqrt{\sum_{k=1}^{l} v_{ik}^2} \sqrt{\sum_{k=1}^{l} v_{jk}^2}}. \tag{7}$$

Classification Accuracy: The classification accuracy (CA) is defined as (8), denoting the accuracy of predicted labels comparing with the labels given in the test data. For (8), T is the document set of test data and $|T|$ is the number of documents in set T. $E(p_i, g_i) = 1$ if $p_i = g_i$ (p_i denotes the predicted label of document i while g_i is the given label in test data corresponding to i), and $E(p_i, g_i) = 0$ if $p_i \neq g_i$.

$$CA = \frac{\sum_{i \in T} E(p_i, g_i)}{|T|} \tag{8}$$

Train & Test: The training and testing process all precompute the cosine similarity of documents using (7). Next, a similarity matrix is as input for nearest-neighbor searching. After the training step, the test data are all labeled with the trained model. Then the CA is obtained using (8).

We use the following seven text representation methods to compare the performance of the AEBoW model.

BoW: The BoW model is described in section I.

LSA: Latent Semantic Analysis [10] is a method to reduce the dimensionality based on BoW.

LDA: Latent Dirichlet Allocation [39].

Net-Local: A complex network method for text classification [26]. We label this method as Net-local, where "local" denotes the local strategy. We only choose the local strategy because the global strategy performs weakly in the experiments, which may be due to that the dimensionality of the representation vector is too low for big datasets.

AE: The average embedding for text representation [15]. AE represents a document as the average of all embeddings of words in the document.

FBoW & FBoWC: FBoW is a fuzzy bag-of-words model [15], which conducts a fuzzy matching through word embeddings. This method is a word embedding based method. FBoWC is an extension of FBoW, which matches the clusters of word embeddings instead.

The word embedding based methods, including AE, FBoW, and FBoWC use the data that are not lemmatized because the learning of word embedding can distinguish all word types. The other methods will use the data after lemmatization.

The implementation of all the methods mentioned above is based on *Python 3.7* with *windows 10* environment. The configuration of the machine we used is *Inter® Core™ i7-8565U CPU @ 1.80GHz; Memory 16.0 GB*. LSA, LDA, and BoW are based on *sklearn* module. The word embeddings of AE are looked up from the pre-trained word embedding dictionary [40], and the words that not in the word embedding dictionary are discarded. AEBoW, FBoW, FBoWC, and Net-local method all run with multi-threads within the permission of the memory.

The dimensionality of representation vectors is set to 3000 for AEBoW, BoW, LSA, LDA, FBoW, and FBoWC. For the Net model, because the number of chosen properties is 8, we set the dimensionality of each property to 3000. So the concatenated vector has a dimensionality of 24000. The vector that projected from AE has dimensionality equal to the word embedding, which is set to 300 in this paper.

Note that we search the best k of KNN for each method, and the searching range is {3, 6, 9, 12, 15, 18, 21, 24, 27, 30}.

C. PERFORMANCE ANALYSIS

Based on the properties of the complex network, including the Degree (D), Eccentricity (E), PageRank (P), and Accessibility (A), we analyze the performance of the AEBoW model. The classification accuracy (CA) is obtained from the dynamic network (co-occurrence network), static network (static semantic network), and hybrid network, respectively. Then the best result for each property is selected. The obtained CA is shown in table 2.

TABLE 2. Classification accuracy (CA).

Datasets	20NG	WebKB	R52	Amazon
BoW	0.5333	0.7529	0.8723	0.6965
AEBoW(D)	0.5402	0.8001	0.8812	0.7091
AEBoW(E)	**0.6900**	**0.8625**	0.8785	0.7702
AEBoW(P)	0.5677	0.8259	**0.8886**	0.7399
AEBoW(A)	0.6010	0.7872	0.8832	0.7270
LSA	0.5567	0.7514	0.8695	0.6905
LDA	0.6773	0.7486	0.8606	0.6668
AE	0.6322	0.7235	0.8586	0.7247
Net-local	0.6263	0.8059	0.8828	0.7301
FBOW	0.6404	0.7958	0.8820	0.7684
FBOWC(mean)	0.6529	0.7994	0.8824	0.7696
FBOWC(max)	0.6508	0.7958	0.8836	0.7672
FBOWC(min)	0.6513	0.7958	0.8851	**0.7708**

Note: The **bold number** denotes the best performance. The proposed model is labeled with *italics*. Mean, max, and min of FBOWC are the way to measure the similarity between word and embedding clusters.

With the same environment, we also get the running time of every method. The results are listed in table 3. Note that the time costs are only counted for the vector projecting process, i.e., the counted period is after the data preprocessing and before the classification.

First, we can observe that the BoW model is the fastest method, though the CA is relatively low. The increase of performance by other methods shows that it is needed to scarify the time for accuracy. The other methods all considerably increase the time costs of text representation while increasing the performance. AEBoW gets the highest CA in 20Newsgroups, WebKB, and Reuters 52, while FBOWC gets the highest CA in Amazon Reviews.

TABLE 3. Time costs (s).

Datasets	20NG	WebKB	R52	Amazon
BoW	2.81	0.63	0.78	0.54
D-AEBoW(D)	24.10	4.22	4.58	4.89
D-AEBoW(E)	**327.16**	73.19	28.94	14.70
D-AEBoW(P)	120.24	22.69	**29.33**	26.91
D-AEBoW(A)	58.58	26.33	5.24	4.89
S-AEBoW(D)	49.46	7.81	12.42	14.01
S-AEBoW(E)	353.70	71.09	40.04	25.04
S-AEBoW(P)	132.99	25.86	35.24	33.63
S-AEBoW(A)	63.14	15.11	14.21	14.77
H-AEBoW(D)	43.36	7.41	13.35	13.57
H-AEBoW(E)	376.33	**95.38**	42.05	24.17
H-AEBoW(P)	140.43	26.44	35.91	33.88
H-AEBoW(A)	99.48	66.71	14.36	15.87
LSA	198.94	36.22	73.80	78.52
LDA	3320.49	596.56	925.45	715.27
AE	185.39	52.62	77.19	84.09
Net-local	3047.33	2641.66	182.47	94.40
FBOW	699.53	129.23	183.14	189.55
FBOWC-c	6628.25	430.96	1404.70	2059.99
FBOWC(mean)	11167.02	770.73	1547.98	1724.99
FBOWC(max)	10990.74	1054.02	1346.54	1693.34
FBOWC(min)	10586.42	1320.73	1360.34	**1703.08**

Note: The **bold number** denotes the time costs of the best performing method. The proposed model is labeled with *italics*. Mean, max, and min of FBOWC are the way to measure the similarity between word and embedding clusters. FBOWC-c means the time costs of k-means clustering operation on word embeddings. The meaning of the prefix for AEBoW, D-dynamic network; S-static network; H-hybrid network.

LSA, LDA, FBOW, and FBOWC are all dimensionality reduction methods. Among these methods, LSA has the lowest time consumption, the accuracy, however, is not competitive. LDA is an iterative approach, the time cost of which is counted within 100 iterations. It can be observed that the CA of LDA can outperform LSA on specific datasets, though the time consumption is always much higher than LSA. The FBOW model and FBOWC model get better accuracy than LSA and LDA. Though FBOWC is better than FBOW, the increase in CA is not acceptable when considering the sharp increase in time cost. The time consumption of FBOWC includes two parts. The first part is the operation of k-means clustering (FBOWC-c in table 3), which rapidly increases following the explosion of the number of vocabulary in the datasets. The second part is the similarity counting between the clusters and the words of a document (mean, max, and

min in table 3). This process makes the similarity calculating repeat thousands (the number of word embedding clusters) of times more than FBOW in small batches, which causes the increase in time costs. Note that the time costs of FBOWC are counted in cases that four threads are used (other methods use eight threads) to avoid out of memory.

AE, FBOW, and FBOWC are word embedding based methods, which all use the pre-trained word embedding during word matching. AE is a simple application on word embedding, which represents a document by simply summing up the embeddings of words in the document. The simple operation loses much high-level information. The results show that, in some cases, the CA of AE is worse than BoW.

AEBoW and Net-local are network-based methods. The main difference between AEBoW and Net-local is that AEBoW uses the individual property as features and uses the BoW idea to collect them. In contrast, Net-local uses different properties that reflect the symmetry of the network and concatenates them as features. Net-local can be seen as the particular case of AEBoW when several properties are chosen, and the top-k features are concatenated. However, using too many local properties can not always improve the performance of text classification while reducing the efficiency on the contrary. The results show that the CA of AEBoW is better than Net-local, and the time costs of AEBoW are much smaller than Net-local.

AEBoW, FBOW, and FBOWC are based on the BoW model. The differences exist that AEBoW is still the sparse representation like BoW, while FBOW and FBOWC solve this limitation by fuzzy matching. However, from the results, we see that the dense representation may not always entirely reflect the right discriminative information for text classification. On the other hand, the dense representation only shows its advantage when using it for dimensionality reduction. If the dimensionality is set to equal in experiments, the sparse characteristic can reduce memory consumption by converting the representation into sparse form (In python 3.7, we can use scipy.sparse module). In contrast, the dense representation can not use specific tools to reduce memory needs. We also tried to use lower dimensionality for FBOW and FBOWC, but this will cause performance reduction. We can also observe that FBOW and FBOWC need more time to process data than AEBoW. We can ascribe it to the difference in matching approach. The properties of the network model will be calculated through matching the words only contains a document, which sometimes only need to match the neighbors, e.g., Degree, Accessibility. On the contrary, FBOW needs to calculate the similarity between each word in a document and all basic terms. Because the basic terms always contain words much more than a document, the time costs are much higher than AEBoW.

D. COMPARISON AMONG THREE KINDS OF NETWORKS

Next, we compare the performance of AEBoW based on three kinds of networks. Figure 6 lists the CA of four datasets.

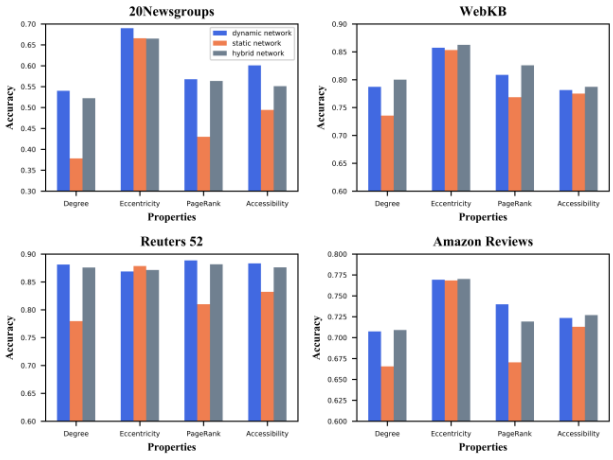

FIGURE 6. CA of three kinds of networks based on Degree, Eccentricity, PageRank, and Accessibility.

First, figure 6 shows that the Eccentricity property can always perform well in all the datasets. It is the only property that produces high CA on three kinds of networks. The other properties all have poor behavior on the static network. We can also observe that the hybrid network can perform a little better on WebKB and Amazon Reviews datasets, which indicates that the combination with relations in both static network and dynamic network can improve the performance of AEBoW in some instances. However, there is no such thing as a free lunch. The hybrid network can not always perform the best.

As table 3 shows, AEBoW on the dynamic network has the best efficiency compared with the hybrid network and network. At the same time, the dynamic network produces competitive results in all four datasets. So the dynamic network is more suitable than the static network and the hybrid network for the AEBoW model.

E. THE INFLUENCE OF K OF KNN IN TEXT CLASSIFICATION

Figure 7 shows the CA of every method in the searching range of k. The results are obtained from the WebKB dataset.

As is shown in figure 7, the accuracy reaches the best in different k for each method, which is the reason that we adopt a searching range of k to select the best results. Most methods reach the best performance when k is around 15, while AEBoW is an exception. The Eccentricity gets the highest CA at $k = 21$.

From figure 7, we can also observe that the results of LSA and BoW are nearly in the same trends, which indicates that LSA is the linearity mapping of BoW with a dimensionality reduction approach. Among the four dimensionality reduction methods (LSA, LDA, FBOW, FBOWC), only FBOW and FBOWC get a satisfactory improvement compared with BoW.

The accuracy of the Eccentricity keeps the best among three kinds of text networks, and the PageRank follows. The results show that some features in AEBoW have a steady

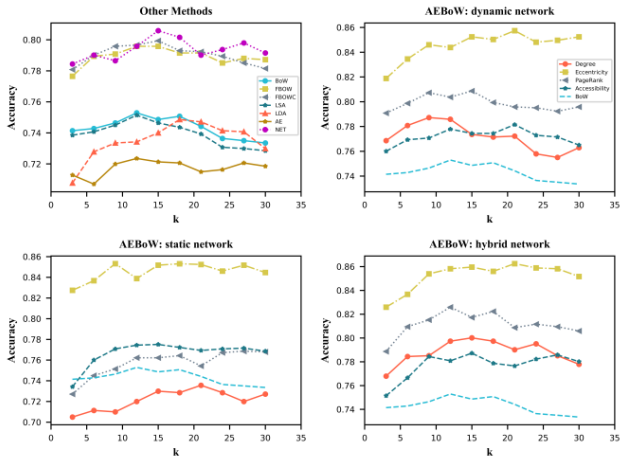

FIGURE 7. Accuracy varying with k of KNN (based on WebKB dataset).

performance despite the kind of networks, and the correlation of words based on the network model can reflect more information than that not based on the network model in text classification. The Degree, Accessibility properties are all local structural properties, and the CA of them is relatively low, indicating that the high-level information of words needs a non-local strategy to extract.

F. HOW DOES AEBOW WORK

The experimental results show that the AEBoW model could outperform the BoW model in specific tasks. In this section, we will discuss part of the reason that the properties of the complex network can perform better.

In the complex network, the nodes affect each other through their links between each other. Even two nodes that are not directed linked can get the influence from the other side through a particular path. The addition and deletion of an edge in the complex network will affect a series of nodes. This character makes the complex network have the ability to capture text structure and semantic change in various ways, and therefore suitable for processing text data.

TABLE 4. Sentence representation of BoW and AEBoW.

Basic terms	{cat, eat, small, white, mouse}	
Sentence	**Sen 1**	**Sen 2**
BoW	[1 1 1 1 1]	[1 1 1 1 1]
Deg	[1 2 2 2 1]	[1 2 1 2 2]
Ecc	[4 3 2 3 4]	[4 3 4 3 2]
Pag ($\times 10^{-2}$)	[13 25 24 25 13]	[13 25 13 25 24]
Acc	[1 1 2 1 1]	[1 1 1 1 2]

To further explain this characteristic without complex math symbols, we use **Sen 1** and **Sen 2** mentioned in section I as a toy example. Different vector forms of these two sentences are listed in table 4. With the BoW model, one can get the same vector to represent the two sentences because there are the same words in the basic terms. However, two sentences contain the opposite meaning. For the AEBoW model, four properties of the complex network all capture the difference between the two sentences.

G. COMPARISON WITH DEEP LEARNING ALGORITHM

The above experiments are all based on the KNN. Next, we also compare the performance of AEBoW with the word embedding model based on the deep learning algorithm. The deep learning algorithm is deployed on *TensorFlow 2.0*.

TABLE 5. Deep learning structure of the word embedding and AEBoW.

AEBoW	Word Embedding
Input Vectors produced by AEBoW	Raw text data after padding
Dense 1	Embedding Layer
(out filters: 300; relu)	(word dimensionality: 300; relu)
Dense 2	Conv1d Layer
(out filters: 200; relu)	(out filters: 300; filter size: 5)
	Maxpool1D Layer
Dense 3	Flatten Layer
(out filters: determined by datasets)	Dense Layer
	(out filters: determined by datasets)

In this experiment, the AEBoW and word embedding model are all applied with the deep learning algorithm. Note that we use different deep learning algorithms for two models because the word embedding model has the corresponding algorithm in deep learning [12] that AEBoW does not fit. The structure of the deep learning algorithm for the two models is listed in table 5. For AEBoW, the inputs are the vectors, and three dense layers are followed. Dense layer 1 and dense layer 2 activate the outputs with Rectified Linear Unit (relu). For the word embedding model, the inputs are the documents after labeling and padding (symbolize the words and pad all documents to the same length). The embedding layer will transfer each word to a vector, the dimensionality of which is 300. The outputs of the embedding layer are convoluted by 1D convolution layer, of which the filter size is 5. The convolution layer will produce 300 filters with relu activation, and the max-pooling layer downsamples the outputs. After downsampling and flattening, the dense layer is used for classification. Note that the dense layer (except the output layer) and the convolution layer all use the biases. Dropout is used before the output layer with a rate of 0.5.

We use the Adam optimization algorithm to update the parameters with mini-batch set to 32, and the learning rate is set to 1e-03. The training epochs is set to 5, and 10% of the training data are selected for cross-validation. The results are listed in table 6. The AEBoW model is based on the dynamic network.

The main part of time costs is different for AEBoW and the word embedding model. For AEBoW, projecting vectors is before training deep learning models. On the contrary, the two steps are finished simultaneous for the word embedding model. Thus the training for AEBoW is much faster than

TABLE 6. CA of AEBoW and the word embedding model.

Datasets	20NG	WebKB	R52	Amazon
Word Embedding	0.6878	0.8847	0.8637	0.8056
AEBoW(D)	0.7703	0.9126	**0.9089**	**0.8263**
AEBoW(E)	0.7691	**0.9169**	0.8917	0.8212
AEBoW(P)	**0.7718**	0.9133	0.9015	0.8260
AEBoW(A)	0.7367	0.8868	0.8921	0.8107

Note: The **bold number** denotes the best performance. The proposed model is labeled with *italics*.

TABLE 7. Time costs (s) of AEBoW and the word embedding model.

Datasets	20NG	WebKB	R52	Amazon
Word Embedding	2859.88	1989.09	218.84	157.41
AEBoW(D)	42.62	13.05	**17.12**	**15.99**
AEBoW(E)	384.08	**92.52**	45.67	30.34
AEBoW(P)	**149.93**	35.23	44.13	34.07
AEBoW(A)	77.53	41.85	18.91	10.91

Note: The **bold number** denotes the time costs of the best performing method. The proposed model is labeled with *italics*.

the word embedding model. We list the time costs of both methods in table 7.

From table 6 and table 7, we can observe that the AEBoW model outperforms the word embedding model on all the four datasets. At the same time, the time costs of AEBoW are much smaller than the word embedding model. However, the word embedding model can accelerate its speed by running on more powerful GPUs while AEBoW can not. The results further certify that the AEBoW can capture more information from text data.

V. DISCUSSION

From the experimental results, it can be observed that the AEBoW model gets good results with high efficiency in text classification. We believe that the application of AEBoW will not only limited to text classification. There are some possible application scenarios of this model, including text interpretation, text clustering, text summarization, and identification of authorship. Next, we briefly describe each application. Furthermore, we also give some ideas for text interpretation.

Text Interpretation is the process of extracting high-level semantics from the raw text data. The high-level semantics are the structured indexes for the raw text data.

Text Clustering is an unsupervised method of machine learning to cluster the documents with high similarity into categories. The AEBoW outputs can be directly used for clustering.

Text Summarization is to catch the key phrase of a document. The key phrase is always a bunch of words from the original document with complete syntax and content.

Identification of Authorship. Each author has his (her) style in their work. The author's style is reflected in the structure, words, or tone of his (her) work. The high-level information can be captured through the AEBoW model.

The following are some ideas about applying AEBoW on text interpretation.

The text interpretation includes processing the unstructured text and extracting the high-level semantics. For the first step, the computer will interpreter a free text correctly into the surface-level form. The free text is analyzed through its syntactic structure, lexical meaning, and then the subsequent computation will take place. By using AEBoW, the surface-level of raw text data can be preprocessed with a network tool, and AEBoW is applied to obtain extra structural and semantic information. For the second step, a series of indexes and complicated relations are derived from the surface-level information. The network model may further explain the patterns of the surface-level information, and the AEBoW model will produce the inputs of the instances object model, which maps the patterns from the surface-level meaning into high-level instance assertions.

It should be noted that the AEBoW model is only a complement to existing methods of text interpretation because there are limitations for AEBoW in grammar parsing and abduction. The AEBoW will not capture the grammars and proper word meaning. So it is needed to introduce the grammar parser and background knowledge.

The AEBoW model is a powerful network-based tool for text analysis, which are possible to be applied to different application scenarios. The introduction of the network model makes AEBoW capture high-level structural and semantic meaning of the text. The application of AEBoW may also need other state-of-the-art studies for a complement.

VI. CONCLUSION

In this paper, we have proposed the AEBoW model based on the complex network to represent text. The AEBoW is an improvement on the BoW model, taking the correlation of words reflected in the text network into consideration. The structure of a text network varies when the different relations of words that form an edge are considered. We have introduced the dynamic network (co-occurrence network) and static network (static semantic network). We have also proposed the hybrid network that contains relations in both the dynamic network and the static network. We have compared the performance of AEBoW with seven text representation methods in text classification.

Experimental results revealed that the proposed AEBoW could get the best performance with high efficiency. The best feature in AEBoW was the Eccentricity, which is a shortest-path-based property of text network. Further analysis showed that for most methods, the performance reaches the best when k is around 15 with KNN as the classifier. For the Eccentricity of AEBoW, the best accuracy exists at $k = 21$. The comparison of the three kinds of networks showed that the dynamic network is more suitable for text classification.

We have also investigated the performance of AEBoW in the deep learning algorithm. By comparing it with the

word embedding model, we certified the high efficiency and excellent performance of AEBoW.

The application of AEBoW is not limited to text classification. Future investigations will be concentrated on using the AEBoW in more text analysis, e.g., text interpretation, text clustering, text summarization and identification of authorship.

REFERENCES

[1] X. Deng, Y. Li, J. Weng, and J. Zhang, "Feature selection for text classification: A review," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3797–3816, Feb. 2019.

[2] H. Kim, P. Howland, and H. Park, "Dimension reduction in text classification with support vector machines," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 37–53, Jan. 2005.

[3] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," in *Proc. 14th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 1997, pp. 143–151.

[4] M. Lan, C. Lim Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 721–735, Apr. 2009.

[5] B. Trstenjak, S. Mikac, and D. Donko, "KNN with TF-IDF based framework for text categorization," in *Proc. DAAAM*, vol. 69. Zadar, Croatia: Univ Zadar, Oct. 2014, pp. 1356–1364.

[6] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec," *Inf. Sci.*, vol. 477, pp. 15–29, Mar. 2019.

[7] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "Critical phenomena in complex networks," *Rev. Modern Phys.*, vol. 80, no. 4, pp. 1275–1335, Oct. 2008.

[8] D. R. Amancio, S. M. Aluisio, O. N. Oliveira, and L. D. F. Costa, "Complex networks analysis of language complexity," *Europhys. Lett.*, vol. 100, no. 5, p. 58002, Dec. 2012.

[9] D. R. Amancio, F. N. Silva, and L. D. F. Costa, "Concentric network symmetry grasps authors' styles in word adjacency networks," *Europhys. Lett.*, vol. 110, no. 6, p. 68001, Jun. 2015.

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, Sep. 1990.

[11] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, no. 1, pp. 177–196, Jan. 2001.

[12] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. (EMNLP)*, Oct. 2014, pp. 1746–1751.

[13] X.-W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014.

[14] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12 pp. 2493–2537, Aug. 2011.

[15] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 794–804, Apr. 2018.

[16] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Phys. Rep.*, vol. 424, nos. 4–5, pp. 175–308, Feb. 2006.

[17] R. F. I. Cancho, and R. V. Solé, "The small world of human language," *Proc. R. Soc. London B* vol. 268, pp. 2261–2265, Nov. 2001.

[18] J. Cong and H. Liu, "Approaching human language with complex networks," *Phys. Life Rev.*, vol. 11, no. 4, pp. 598–618, Dec. 2014.

[19] S. M. G. Caldeira, T. C. P. Lobao, R. F. S. Andrade, A. Neme, and J. G. V. Miranda, "The network of concepts in written texts," *Eur. Phys. J. B*, vol. 49, no. 4, pp. 523–529, Aug. 2005.

[20] L. Antiqueira, M. G. V. Nunes, O. N. Oliveira, Jr., and L. D. F. Costa, "Strong correlations between text quality and complex networks features," *Phys. A, Stat. Mech. Appl.*, vol. 373, pp. 811–820, Jan. 2007.

[21] H. F. de Arruda, L. D. F. Costa, and D. R. Amancio, "Topic segmentation via community detection in complex networks," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 26, no. 6, Jun. 2016, Art. no. 063120.

[22] M. Garg and M. Kumar, "Identifying influential segments from word co-occurrence networks using AHP," *Cognit. Syst. Res.*, vol. 47, pp. 28–41, Jan. 2018.

[23] C. Akimushkin, D. R. Amancio, and O. N. Oliveira, "Text authorship identified using the dynamics of word co-occurrence networks," *PLoS ONE*, vol. 12, no. 1, 2017, Art. no. e0170527.

[24] D. R. Amancio, O. N. Oliveira, Jr., and L. D. F. Costa, "Structure–semantics interplay in complex networks and its effects on the predictability of similarity in texts," *Phys. A, Stat. Mech. Appl.*, vol. 391, no. 18, pp. 4406–4419, Sep. 2012.

[25] L. Antiqueira, O. N. Oliveira, L. D. F. Costa, and M. D. G. V. Nunes, "A complex network approach to text summarization," *Inf. Sci.*, vol. 179, no. 5, pp. 584–599, Feb. 2009.

[26] H. F. de Arruda, L. D. F. Costa, and D. R. Amancio, "Using complex networks for text classification: Discriminating informative and imaginative documents," *Europhys. Lett.*, vol. 113, no. 2, p. 28007, Jan. 2016.

[27] D. R. Amancio, "Probing the topological properties of complex networks modeling short written texts," *PLoS One*, vol. 10, no. 2, 2015, Art. no. e0118394.

[28] D. Yan, K. Li, and J. Ye, "Correlation analysis of short based on network model," *Phys. A, Stat. Mech. Appl.*, vol. 531, Oct. 2019, Art. no. 121728.

[29] H. F. de Arruda, V. Q. Marinho, T. S. Lima, D. R. Amancio, and L. D. F. Costa, "An image analysis approach to text analytics based on complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 510, pp. 110–120, Nov. 2018.

[30] M. Sigman and G. A. Cecchi, "Global organization of the Wordnet lexicon," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 3, pp. 1742–1747, Feb. 2002.

[31] A. N. Langville and C. D. Meyer, "A survey of eigenvector methods for Web information retrieval," *SIAM Rev.*, vol. 47, no. 1, pp. 135–161, Jan. 2005.

[32] M. Craven, D. Freitag, A. Mccallum, and T. Mitchell, "Learning to extract symbolic knowledge from the World Wide Web," in *A Comprehensive Survey of Text Mining*, M. W. Berry, Ed, Heidelberg, Germany: Springer, 2003.

[33] [Online]. Available: http://ana.cachopo.org/datasets-for-single-label-text-categorization

[34] A. Cardoso-Cachopo, "Improving methods for single-label text categorization," Ph.D. dissertation, Instituto Superior Técnico, Lisboa, Portugal, Oct. 2007.

[35] [Online]. Available: https://gist.github.com/kunalj101

[36] G. D. Gao, H. Wang, D. Bell, Y. X. Bi, and K. Greer, "KNN model-based approach in classification," in *Proc. OTM Int. Conf. CoopIS DOA ODBASE*, Catania, Italy, Nov. 2003, pp. 986–996.

[37] Y.-S. Lin, J.-Y. Jiang, and S.-J. Lee, "A similarity measure for text classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1575–1590, Jul. 2014.

[38] G. F. de Arruda, A. L. Barbieri, P. M. Rodríguez, F. A. Rodrigues, Y. Moreno, and L. D. F. Costa, "Role of centrality for the identification of influential spreaders in complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 90, no. 3, Sep. 2014, Art. no. 032812.

[39] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[40] [Online]. Available: https://nlp.stanford.edu/projects/glove/

DONGYANG YAN was born in Xuchang, Henan, in 1993. He is currently pursuing the Ph.D. degree in system science with the Key Laboratory of Rail Traffic Control and Safety in Beijing Jiaotong University. His main research interest is natural language processing with complex networks and other methods.

KEPING LI is currently a Professor with the State Key Laboratory of Rail Traffic Control and Safety. He was elected in New Century Talent Supporting Project by Education Ministry. His main research interests include modeling of the complex network system and modeling, analysis, and optimization of rail transit systems.

SHUANG GU was born in Yichun, Heilongjiang, in 1994. She is currently pursuing the Ph.D. degree with the Key Laboratory of Rail Traffic Control and Safety in Beijing Jiaotong University. Her main research interest is the complex networks.

LIU YANG was born in Guizhou, in 1990. She received the B.Sc. degree in information and computing science and the master's degree in logistics engineering from Guizhou University. She is currently pursuing the Ph.D. degree with the Key Laboratory of Rail Traffic Control and Safety in Beijing Jiaotong University. Her main research interest is complex network and risk analysis in transportation.