IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 5, 2018, accepted September 26, 2018, date of publication October 1, 2018, date of current version October 29, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2873327

Neural Feedback Text Clustering With BiLSTM-CNN-Kmeans

YANG FAN, LIU GONGSHEN, MENG KUI, AND SUN ZHAOYING
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding authors: Liu Gongshen (lgshen@sjtu.edu.cn) and Meng Kui (mengkui@sjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61472248, Grant 61772337, and Grant U1736207, and in part by the Shanghai Technology Research Leadership Program under Grant 16XD1424400.

ABSTRACT Text clustering is a very important technique in the field of data mining. It is widely used in information retrieval and text integration. Most existing studies focus on the optimization of feature extraction methods or clustering algorithms separately. In this paper, we propose a neural feedback clustering algorithm combining bidirectional long short-term memory and convolutional neural network with kmeans method. Unlike previous papers, the proposed algorithm treats feature extraction and clustering as a united process, where clustering results can be used as feedback information to dynamically optimize the parameters of networks. Experimental results show that the proposed algorithm achieves significant improvements compared with other existing text clustering algorithms and has a certain degree of noise robustness.

INDEX TERMS Neural feedback clustering, feature extraction, text clustering, BiLSTM-CNN-Kmeans network.

I. INTRODUCTION

As the number of Internet users surges every year, text information on the Internet has exploded and network is flooded with vast amounts of textual data. Text clustering technology [1] for massive web content has received widespread attention. Text clustering refers to the process of analyzing a group of texts and classifying similar texts into the same category. As an unsupervised data mining technology, text clustering does not require training the model ahead of time, or making manual annotations of texts beforehand. Compared with other natural language processing (NLP) algorithms like classification, clustering algorithm requires less human intervention and has higher efficiency. Therefore, text clustering has become a key technology in natural language processing [2] and has been widely used in various fields such as information retrieval, organization and processing [3], [4].

In text clustering, feature extraction is an essential process for creating an accurate clustering model. Vector space model (VSM) [5], the term frequency-inverse document frequency (TF-IDF) measure [6], and latent semantic analysis (LSA) [7] are widely used in traditional text clustering to represent documents. However, these methods confronted with high data sparseness and complex semantics, as well as large noise interference. Some researchers extended semantics of words through external links or world knowledge bases, while these methods would suffer from an inefficiency when processing large-scale corpora [8], [9].

In recent years, deep learning has become the hottest area in natural language processing because of its good performance and great potential in implementing artificial intelligence. With the help of word embedding, deep learning models achieve remarkable results for text representation. Recurrent neural network (RNN) is able to learn long-term dependencies of texts rather than local features. Tai *et al.* [10] introduce a long short-term memory (LSTM) model to improve the semantic representations. Convolutional neural network (CNN) models have been shown to be effective for extracting local features. With the utilization of CNN, significant improvement has been achieved in the field of computer vision tasks [11]–[17]. For natural language processing, CNN also shows its great potentials. Santos and Gatti [18] propose a CNN model for sentiment analysis. A CNN based structure is used by Xu *et al.* [19], [20] for text clustering. Therefore, a proper combination of RNN(or LSTM) and CNN structure could benefit from the advantages of both structures.

Moreover, feature extraction and clustering process are normally executed separately. Many researches tend to focus on either feature extraction approaches [5], [19], [20] or the improvement of clustering algorithms [21]. However, clustering can be used in neural network processing to adjust parameters dynamically for better performance. In view of

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

mentioned difficulties and challenges, we propose a neural feedback text clustering algorithm based on BiLSTM-CNN-Kmeans network (LCK-NFC algorithm hereafter). The main contributions of this paper are summarized as follows:

1) A feature extraction method based on bidirectional LSTM-CNN layer is presented. Firstly, based on a large-scale corpus, this method uses word2vec model to train the corpus, learns the semantic relation between two words, uses word vectors to represent words, and transforms given texts into sparse original vector forms. Then this method expands word representation based on pre-trained word embedding, and obtains the context semantics of words in the text through a BiLSTM layer. Finally, a CNN layer is utilized to extract local features of texts.
2) We firstly propose a neural feedback text clustering algorithm by incorporating clustering procedure into neural network, so that clustering results can be used as feedback information to dynamically optimize network parameters. After extracting features of texts, k-means algorithm is utilized to execute text clustering, and inverted Silhouette Coefficient of clustering result is selected as the loss function of our neural network. Therefore, the feature extraction and the text clustering is continuously optimized during training procedure.

The remainder of this paper is organized as follows: Section II introduces some related works. Section III illustrates the proposed methodology in details. Section IV shows the experiment results and discussions. Conclusion and future work are provided in the last Section.

II. RELATED WORKS

Traditional methods of text clustering include the VSM model, TF-IDF measure, and the LSA model [5]–[7]. A generalized VSM model presented by Seifzadeh *et al.* [5], which is able to utilize the relationships between pairs of terms so that compensate for the sparsity of short texts. Term frequency-inverse document frequency (TF-IDF) is a numerical statistical method, which is used to evaluate the relevance of a word to a particular text in a text collection. The importance of a word is positively related to its frequency of occurrence in the text and negatively related to its frequency of occurrence throughout the text collection. Firstly, it computes the normalized term frequency and the inverse document frequency to form the tf-idf weight, then text vector can be generated based on the tf-idf weight of texts. Neto *et al.* [6] use TF-IDF measures for text clustering and summarization. Latent semantic analysis (LSA) [7] is used to extract and represent the contextual-usage meaning of words by statistical computations. They use singular value decomposition (SVD) to reduce large sparse vector spaces for word meaning, and then word embeddings is generated from term-documents matrix. In addition, some sophisticated models are incorporated to improve the performance.

Some researchers focus on exploring the sophisticated models for short text clustering. Tang *et al.* [22] propose a framework that uses matrix factorization methods to perform multi-language knowledge integration and feature reduction simultaneously. Yin and Wang [23] illustrate a collapsed Gibbs Sampling algorithm, and use Dirichlet Multinomial Mixture model for short text clustering tasks. Wazarkar *et al.* [24] implement text clustering by combining both hierarchical clustering and fuzzy clustering, and the proposed HFRECCA algorithm is effective for capturing the hierarchical information of texts. Song *et al.* [25] introduce a hybrid evolutionary computation method to optimize the centers searching process of clustering. Thus, the clustering problem can be simulated by using the evolutionary process of chromosomes or particles encoded by centers ways. To overcome the sparsity and high-dimensionality of text documents, Jia *et al.* [26] propose WordCom concept decomposition method. They use the community detection method k-rank-D to extract concept vectors from word co-occurrence network. Experimental results show that WordCom algorithm is robust to sparse short texts.

With the development of word embedding technologies, remarkable performances of feature extraction have been achieved in many tasks of natural language processing (NLP). There are series of models have been proposed for sentences or documents representation. Mikolov *et al.* [27] propose a word2vec model, which has the ability of capturing both syntactic and semantic information of texts. The basic idea of word2vec is to use both skip-gram and continuous bag-of-words models to train the large datasets, and then obtain the final representation of words or characters. Pennington *et al.* [28] introduce a GloVe model for word representation, where word vectors are trained on aggregated global word-word co-occurrence statistics from a corpus.

CNN has the ability of local features extraction, which could largely reduce the complexity of networks. In many computer vision tasks, CNN is preferred to be able to extract features. Han *et al.* [12] apply metric learning to co-saliency detection and adopt the off-the-shelf CNN for initial feature extraction. Gheng *et al.* [14] conduct experiments on deep CNN features to evaluate the proposed DML method. For category-specific object detection task, deep CNN-based models have the ability to extract rich features [13]. Robust object co-segmentation architecture is implemented by Han *et al.* [15], which introduces "CNN-S" model to explore higher-level semantic features. In addition, discriminative CNN features is ultilized to improve the performance of remote sensing image scene classification [16]. CNN also can be used to learn features from words or texts. Xu *et al.* [19], [20] proposed a flexible CNN framework for short text representation.

RNN is used to learn the sequential information of texts. Some researchers focus on the combination of CNN and RNN (or LSTM) in solving NLP tasks. Wang *et al.* [29] combine CNN and RNN together for sentimental analysis of short texts. The proposed architecture takes both coarse-grained local features and long-distance dependencies into consideration, which achieves outstanding performance for

sentimental analysis tasks. Wu *et al.* [30] present a CNN-LSTM with attention model to predict emoji. CNN-LSTM is used to learn local and long-range contextual information, and attention strategy is used to select important components. Song *et al.* [31] propose an LSTM-CNN based framework to explore fine-grained semantic phrases for abstractive text summarization.

Recently, some researchers use neural network for text clustering. Xu *et al.* [19] firstly apply CNN and some dimensionality reduction methods for feature extraction of short texts. Then, they enhance the structure to couple various semantic features in [20]. In this algorithm, pre-trained word vectors are firstly embedded into compact binary codes, then CNN is applied to learn deep feature representation. Finally, clusters are obtained by employing k-means algorithm. Combine LSTM and CNN together is an efficient way to extract contextual semantic information. In this paper, we firstly use BiLSTM-CNN model to extract features for text clustering, and propose an unsupervised neural feedback text clustering algorithm, without involving any hand-craft or knowledge based methods.

III. PROPOSED METHOD

A. OVERVIEW OF PROPOSED LCK-NFC ALGORITHM

The flow chart of the proposed LCK-NFC algorithm is shown in Fig. 1. Given the raw texts collection, the goal is to cluster these texts into different clusters based on extracted features. This algorithm consists of two main parts, text preprocessing and neural feedback clustering. Text preprocessing is the conversion of raw, unstructured text data into a proper format, including tokenization (or word segmentation for Chinese) [32], [33] and stop word removal. The sparse vector form of text can be generated through the vectorization of structured texts, where BiLSTM is used to obtain context semantics. Then, CNN is applied to extract features of original text embedding. Finally, text clustering is implemented by using k-means algorithm. To optimize network structure and access to the best clustering result, LCK-NFC algorithm considers the concentration of clustering results as the loss function of the whole network, in order to jointly optimize parameters for feature extraction and clustering process. The common notations and operators employed in this paper are summarized in Table 1.

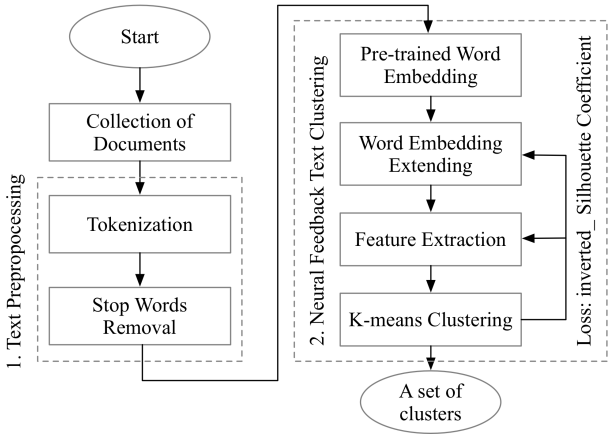


FIGURE 1. The flow chart of proposed LCK-NFC algorithm.

TABLE 1. List of notation.

Notation	Description
m	the number of texts
n	the number of words in one text
x	input word embeddings
x_i	i-th word embedding
$x_{i:j}$	the collection of $x_i, x_{i+1}, ..., x_j$
c	feature maps
c_i	the feature map of i-th word
$\hat{c}$	max-pooling results of c
W	weight matrix of convolutional filters
b	bias matrix of convolutional filters
h	the height of window size
z	the convolutional results
z'	the dropout results
r	binary vectors with Bernoulli distribution
p	dropout probability
w_i	i-th word
$e(w_i)$	i-th word embedding
$h_l(w_i)$	i-th output vector of forward LSTM layer
$h_r(w_i)$	i-th output vector of backward LSTM layer
W^l	weight matrix of forward LSTM hidden layer
W^r	weight matrix of backward LSTM hidden layer
W^{el}	weight matrix of forward LSTM input layer
W^{er}	weight matrix of backward LSTM input layer
b_l	bias matrix of forward LSTM hidden layer
b_r	bias matrix of backward LSTM hidden layer
$\hat{x_i}$	extended i-th word embedding
SC	Silhouette Coefficient
s_i	the Silhouette Coefficient value of i-th text
b_i	internal cluster distance of i-th text
a_i	external cluster distance of i-th text
K_i	category of i-th text
C_j	cluster number of i-th text
$\|K_i\|$	the number of in the category K_i
$\|C_j\|$	the number of texts clustered into the cluster C_j
n_{ij}	the number of texts in the cluster C_j with category K_i
$F(K_i, C_j)$	F1 score
$P(K_i, C_j)$	precision
$R(K_i, C_j)$	recall
$\|D\|$	the number of experimental texts

B. EVALUATION MATRIX

C. TEXT PREPROCESSING

When dealing with Chinese documents, text preprocessing usually includes Chinese word segmentation (CWS), removal of stop words and punctuations. Word segmentation is the process of dividing consecutive original texts into a collection of characters, words and phrases according to certain rules. There are many previous works designed for CWS. Chang *et al.* [32] optimize CWS task by introducing a CRF model. Wang and Xu [33] present a CNN-CRF model, which applied parallelized CNNs to train the network efficiently and achieve competitive performance for CWS. For convenience of experiment, the CWS system used in this paper is based on NLPIR (also called ICTCLAS 2013).[1] NLPIR for CWS is built with hierarchical hidden markov model, which is

[1] http://ictclas.nlpir.org

an open source tools developed by [34]. Besides, Wikipedia entries are added to thesaurus of the system, since the accuracy of word segmentation degrades as new words appear.

Stop words refer to words without a real meaning, which have no contribution or even have a negative effect on text categorization during textual analysis. In fact, both Chinese and English texts are filled with a large number of meaningless stop words, which account for a great proportion of texts. In order to avoid feature vectors redundancy as much as possible, stop words should be removed before further analysis and processing of the text.

There are basically two ways to construct a stop words list, one is a rule-based construction method, the other is an automatic learning method based on statistical analysis. There are many well-known stop words lists in English, such as the stop words lists presented in [35] and [36]. Compared to English, Chinese vocabulary is much larger and Chinese words tend to be ambiguous. In this paper, we combine Baidu list of stop words, HIT stop list and Sichuan University Machine Intelligence Laboratory disabled thesaurus and remove duplicate words to form the stop word list.[2]

D. CONVENTIONAL CNN FOR FEATURE EXTRACTION

Convolution neural network (CNN) was first proposed by LeCun and Bengio [37] in the field of handwritten recognition. In recent years, CNN has good performance and breakthrough in language modeling [18], [19], [38]. Fig 2 illustrates conventional CNN model for feature extraction, which is based on the model proposed by Kim [38].

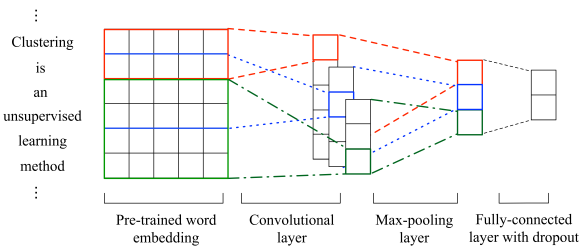


FIGURE 2. The structure of conventional CNN model.

This model first trains word2vec model on a large Chinese Wikipedia corpus,[3] learns semantic relation between two words, uses word vector (or word embedding) $x_i \in R^k$ to express words, where k is the dimension of word vector. So, x_i is a k-dimensional vector corresponding to the ith word in the text. For a text with length n is represented as:

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \tag{1}$$

where $\oplus$ is a connection operator, and $x_{i:j}$ denotes a connection of $x_i, x_{i+1}, \ldots, x_j$.

The input of the network is an $m \times n$ matrix, where m is the number of texts, which contain n words in each text. In order to ensure word integrity, convolution filters constructed in this paper can only slide vertically, and its width must be consistent with the dimension of word vector. The convolution of the input matrix with a filter determined by weights $W \in R^{hk}$ and bias $b \in R$, the feature map c_i is obtained from word combination $x_{i:i+h-1}$ according to the following formula:

$$c_i = f(W \cdot x_{i:i+h-1} + b) \tag{2}$$

where f is an activation function, a common choice is a sigmoid function or other nonlinear functions. By applying this filter to all possible window size with height h:

$$\{x_{1:h}, x_{1:h+1}, \ldots, x_{n-h+1:n}\} \tag{3}$$

As shown in Fig. 2, the convolutional operations with different colors denote different heights of window size, where red, blue and green stand for the value of height with 2, 3 and 4 respectively. Thus, we can obtain a collection of feature maps $c \in R^{n-h+1}$:

$$c = \{c_1, c_2, \ldots, c_{n-h+1}\} \tag{4}$$

Max-over-time pooling can progressively reduce the spatial size of the representation and keep the most important feature $\hat{c} = max\{c_i\}$. So the output of this layer is z:

$$z = \left\{\hat{c_1}, \hat{c_2}, \ldots, \hat{c_m}\right\} \tag{5}$$

In order to reduce over-fitting during network training and improve the accuracy of neural network model, we applied dropout technique [39] in the last full-connected layer. Dropout stochastically sets the activations of hidden units to zero according to a proportion p at training time. The dropout method first generates binary vectors r with the same dimensions of z based on the Bernoulli distribution (elements in the vector can only be 0 or 1):

$$r \sim Bernoulli(p) \tag{6}$$

The output of this model is calculated as follows:

$$z' = z^{\circ} r \tag{7}$$

where $\circ$ is multiplication operation. Backpropagation only calculates gradients for elements that are not discarded. When making final prediction, all the parameters are multiplied by $1 - p$ with actual final network model parameters.

This model gets several different features through a large number of filters with different window sizes. These features constitute the penultimate layer of the network and are passed to a fully-connected layer, output of which is the feature extraction result. Note that a scalar is obtained after convolution and pooling layer, since the width of convolution filter and word vector are the same. Finally, feature vectors of all window sizes are combined into a complete eigenvector as the final text vectorized representation, which are used as input of clustering algorithm.

[2]https://github.com/chdd/weibo/tree/master/stopwords
[3]http://linguatools.org/tools/corpora/wikipediamonolingualcorpora/

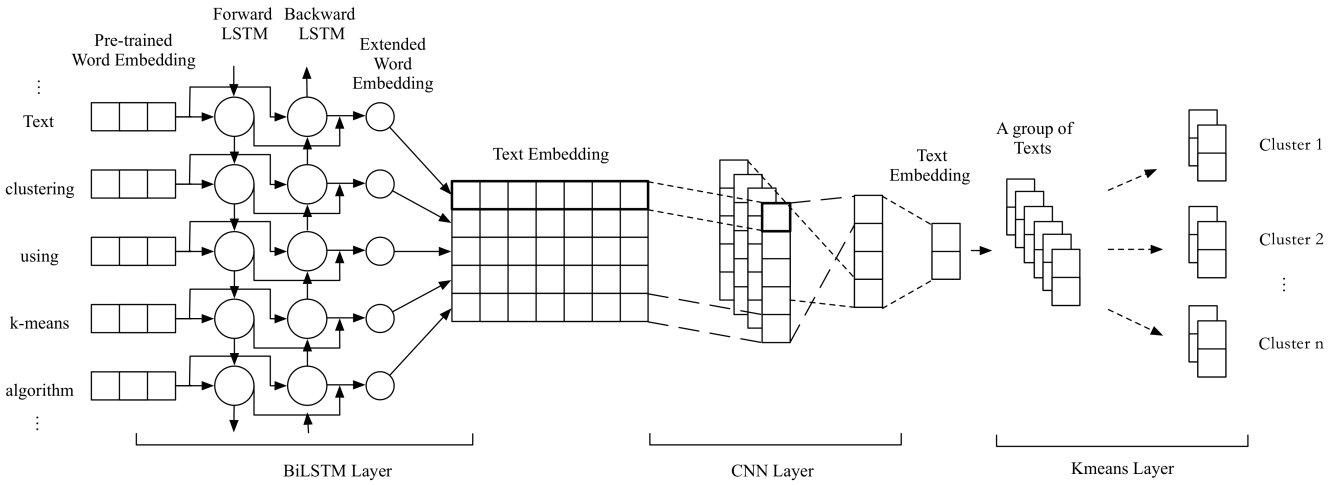


FIGURE 3. The structure of proposed BiLSTM-CNN-Kmeans network.

E. ENHANCED BiLSTM-CNN FOR CONTEXTUAL AND LOCAL FEATURE EXTRACTION

Although CNN has better performance than RNN when capturing the semantics of texts, it is difficult for CNN to decide the window size of the filters. Small window size may result in the loss of important information, whereas large size may result in the explosion of parameters space, then the difficulty of model training is increased. To address above challenges, we add a long short-term memory (LSTM) layer [40] before CNN layer to enhance the learning structure. LSTM is a kind of RNN, which is capable of learning long-term dependencies. In the text, the meaning of each word is related to both the previous text contents and the latter text contents. Therefore, a bidirectional LSTM (BiLSTM) network [41]is applied to learn past features (via forward cells) and future features (via backward cells) at a specific time.

Based on conventional CNN network, a BiLSTM-CNN layer is proposed to grasp text semantics, as shown in the left part of Fig. 3, which illustrates the overall architecture of BiLSTM-CNN-Kmeans network. The input of this network is a context, composed of words sequence $\{w_1, w_2, \ldots, w_n\}$ BiLSTM is used to extend the word embedding, by combining the word itself and its context to represent the word. $h_l(w_i)$ and $h_r(w_i)$ denotes the output vector of forward and backward LSTM layer respectively, which contains the left and right context of word w_i, computed as following formulas:

$$h_l(w_i) = f(W^l h_l(w_{i-1}) + W^{el} e(w_{i-1}) + b_l) \quad (8)$$

$$h_r(w_i) = f(W^r h_r(w_{i+1}) + W^{er} e(w_{i+1}) + b_r) \quad (9)$$

where $e(w_i)$ is the word embedding of word w_i. W^l (or W^r) is a weight matrix that transforms the hidden layer forward (or backward) into next hidden layer. W^{el}(or W^{er}) and is weight matrix of transformation of input word embeddings of forward (or backward) layer. b_l (or b_r) is the bias of forward (or backward) layer. f is a non-linear activation function.

Besides, all the texts in the forward layer share the same vector $h_l(w_1)$, and share vector $h_r(w_n)$ in the backward layer. The time complexity is $O(n)$ for generating all vectors $h_l(w_i)$ and $h_r(w_i)$. As a result, we get a better representation $\hat{x}_i$ of the word, as shown in following formula:

$$\hat{x}_i = h_l(w_i) \oplus h_r(w_i) \quad (10)$$

The window size of convolution filter is fixed to $1 \times k$, where k is the dimension of word embeddings. This setting not only ensures the text integrity, but also reduces the number of convolution kernels and the number of network training parameters. Therefore, the time complexity is reduced.

F. NEURAL FEEDBACK CLUSTERING ALGORITHM

As depicted in Fig. 3, after obtaining semantic representation of texts, we feed final text representation into clustering layer to implement neural feedback text clustering. We apply k-means algorithm based on [42] to perform text clustering.

In order to dynamically adjust and optimize network by the interaction of feature extraction and clustering process, through forward and backward propagation, we define the loss function of neural network as inverted Silhouette Coefficient [43]. By minimizing the loss function, the whole neural network can be adjusted to the optimal structure and the clustering result can be optimal. For a proper clustering solution: texts in the same cluster are highly similar to each other, while in different clusters are highly dissimilar to each other. Silhouette Coefficient is a metric to evaluate the clustering result, and Silhouette Coefficient of the texts can be defined as:

$$SC = \frac{1}{N}\sum_{i=1}^{N} s_i = \frac{1}{N}\sum_{i=1}^{N} \frac{b_i - a_i}{max(a_i, b_i)} \quad (11)$$

where s_i is Silhouette Coefficient of text i, a_i is the average distance to the other texts in the same cluster, and b_i is the average distance to other sample in the next-nearest cluster.

The number of texts is N. The value of SC ranges from -1 to 1, where higher value indicates a more reasonable clustering result. Therefore, loss function is defined as inverted Silhouette Coefficient:

$$LOSS = -SC = -\frac{1}{N}\sum_{i=1}^{N}\frac{b_i - a_i}{max(a_i, b_i)} \tag{12}$$

After the loss function is defined, the feature extraction process of neural network can be continuously trained, adjusted and optimized according to the loss function. Until the loss function is minimized, the clustering result reaches the optimal condition under the existing conditions.

IV. EXPERIMENTS AND DISCUSSIONS

A. DATASET AND PREPROCESSING

1) SOGOUCS[4]

The dataset is collected from Sohu News for 18 channels including learning, sports, health, entertainment and stock during June-July 2012. In this paper, we chose five categories, with a total of 173782 texts as original experimental data, as shown in Table 2. To evaluate the performance on both long text and short text, we utilize news content as long text dataset and news headline as short text dataset.

TABLE 2. Statistics for the SogouCS text datasets.

Category	Number
Learning	11738
Sports	43391
Health	23282
Entertainment	49407
Stock	45964

2) FUDAN CORPUS[5]

Fudan University text classification corpus is collected by Fudan university, including a training set and a testing set. The training set has 9804 texts and testing set has 9833 texts. Both of them are divided into 20 categories, and we select five categories as our experimental datasets, as shown in Table 3.

TABLE 3. Statistics for the Fudan Corpus.

Category	Number
Politics	1026
Computer	1358
Agriculture	1022
Economy	1601
Sports	1254

3) CHNSENTICORP

ChnSentiCorp [44] is a Chinese sentiment corpora, including ChnSentiCorp-Htl-ba-4000 and ChnSentiCorp-NB-ba-4000, which corresponding to the domains hotel and computer. The statistics of ChnSentiCorp is shown in Table 4.

[4]http://www.sogou.com/labs/resource/cs.php
[5]http://www.nlpir.org/?action-viewnews-itemid-103

TABLE 4. Statistics for the ChnSentiCorp.

Category	Number
Hotel	4000
Computer	4000

B. EVALUATION MATRIX

In this experiment, clustering performance is evaluated by comparing clustering results with given tags in the dataset. F1 score (or F-measure) [45] is used as evaluation matrix. For the text with category K_i, and clustered into cluster C_j, F1 score can be interpreted as a weighted average of precision and recall:

$$F(K_i, C_j) = \frac{2 \cdot P(K_i, C_j) \times R(K_i, C_j)}{P(K_i, C_j) + R(K_i, C_j)} \tag{13}$$

where precision $P(K_i, C_j)$ and recall $R(K_i, C_j)$ are defined by:

$$P(K_i, C_j) = \frac{n_{ij}}{|C_j|} \tag{14}$$

$$R(K_i, C_j) = \frac{n_{ij}}{|K_i|} \tag{15}$$

where n_{ij} is the number of texts belonging to cluster C_j in the category K_i, $|C_j|$ is number of texts clustered into the cluster C_j, and $|K_i|$ is the number of texts in category K_i. Therefore, the average F1 score of overall cluster can be calculated as follows:

$$F(C) = \sum_{K_i=1}^{K}\frac{|K_i|}{|D|}max_{C_j \in C}\{F(K_i, C_j)\} \tag{16}$$

where $|D|$ is the number of experimental texts. Due to the random selection of clustering initial points and network parameters, we choose the average value of multiple experiments results as final result of our experiments.

C. EXPERIMENTAL SETTINGS

The parameter settings of our BiLSTM-CNN-Kmeans network is described in Table 5. In the training step, the batch size is fixed to the same as the size of training set, and the loss function is inverted Silhouette Coefficient. The optimizer is Adam Optimizer and the learning rate is set to 0.001. The network parameters of convolutional layer are randomly

TABLE 5. The parameter settings of our BiLSTM-CNN-Kmeans network.

Parameters	Settings
Initial learning rate	0.001
Word embedding dimension	128+64*2
Filter size	1*256
Filter numbers	100
Activation function	ReLU
Pooling	Max-pooling
Dropout rate	0.5
Batch size	100
Optimizer	AdamOptimizer
Clustering algorithm	K-means algorithm

initialized according to the normal distribution, and the initial centers of clustering layer are randomly selected in the training samples.

In this paper, we conduct comparative experiments by using SogouCS corpus to generate long and short text dataset, as introduced in subsection D, E, F. The reason is that each category of Fudan corpus has limited number of texts and ChnSentiCorp has only two categories. In order to verify the effectiveness and universality of the proposed algorithm, we compare the performance on three difference corpora, as illustrated in subsection G. Finally, we show the most important words extracted by proposed LCK-NFC algorithm.

D. EVALUATION OF PROPOSED LCK-NFC ALGORITHM

To explore the performance of proposed BiLSTM-CNN layer and feedback clustering layer on text clustering task, we implement complete LCK-NFC algorithm and partially-deleted LCK-NFC algorithms on short text and long text datasets. Both the number of iteration and convolutional kernel are set to 100.

TABLE 6. Evaluation of the proposed LCK-NFC algorithm on SogouCS long text dataset in F1 scores (%).

Number	LCK-NFC (without BiLSTM)	LCK-NFC (without Feedback)	LCK-NFC
1000	68.1	67.3	70.0
2000	72.7	71.5	73.6
3000	79.4	76.2	82.4
AVG	73.4	71.6	**75.3**

TABLE 7. Evaluation of the proposed LCK-NFC algorithm on SogouCS short text dataset in F1 scores (%).

Number	LCK-NFC (without BiLSTM)	LCK-NFC (without Feedback)	LCK-NFC
10000	72.3	69.8	74.5
20000	78.0	73.2	82.8
30000	79.7	75.7	83.3
AVG	76.7	72.9	**80.2**

Experimental results are given in Table 6 and Table 7. Experimental results show that the larger the training set is, the better the performance of LCK-NFC algorithm can be. Moreover, different optimization structures in LCK-NFC algorithm have different effects on overall performance improvement of the algorithm.

If we remove feedback clustering part of LCK-NFC algorithm and only retain BiLSTM-CNN layer, the improvement of clustering performance is not obvious. Although the extended word embedding contains more complete information than the pre-trained word embedding, the promotion on feature extraction process and the text clustering process is limited. On the other hand, if BiLSTM layer in LCK-NFC algorithm is removed and other optimization structures are retained, the clustering effect of the algorithm will not be greatly affected. It shows that the feedback clustering part of LCK-NFC algorithm is the most important part of improving the clustering effect.

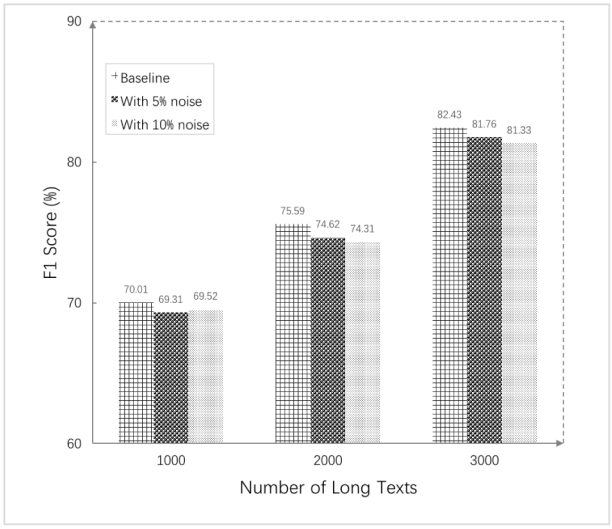


FIGURE 4. Noise robustness of proposed LCK-NFC algorithm on SogouCS long text dataset.

E. NOISE ROBUSTNESS

In text clustering, the performance of clustering algorithms tends to be disrupted with noise data introduced. This problem can be noted as a noise robustness issue. To examine the robustness characteristics of clustering algorithms to noise, the comparative experiments have been carried out with different proportions of noise data, as shown in Fig. 4 and Fig. 5. Noise data is selected from other categories of texts in SogouCS, such as Technology, Lifestyle, Politics and so on. All the experiments were implemented with the same number of clustering texts, with additional noise data in 0%, 5%, 10%

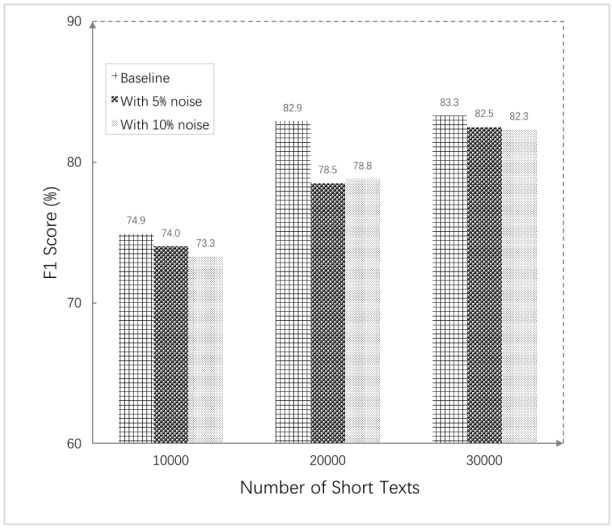


FIGURE 5. Noise robustness of proposed LCK-NFC algorithm on SogouCS short text dataset.

TABLE 8. Lists of important words on SogouCS.

Categories	Words				
Learning	College	Batch	Average Score	Professionals	College Entrance Examination
教育	高校	批次	平均分	专业	高考
Sports	Sports	Sports	Training	Exercise	Physical Fitness
体育	体育	运动	训练	锻炼	体质
Health	Disease	Hospital	Treatment	Outpatient	Patients
健康	疾病	医院	治疗	门诊	患者
Entertainment	Entertainment	Film	Ceremony	TV series	Shows
娱乐	娱乐	影视作品	盛典	电视剧	节目
Finance	Economics	Finance	Securities	Stocks	Company
财经	经济	财政	证券	股票	公司

TABLE 9. Lists of important words on Fudan corpus.

Categories	Words				
Politics	Policy	Politics	Systems	Government	Society
政治	政策	政治	制度	政府	社会
Computer	Computer	Programs	Network	Software	Distributed
计算机	计算机	程序	网络	软件	分布式
Agriculture	Agriculture	Crop	Cultivated land	Rural	Farmer
农业	农业	农作物	耕地	农村	农民
Economy	Economy	Consumer	Market	Capital	Finance
经济	经济	消费	市场	资本	金融
Sports	Sports	Training	Fitness	Exercise	Physical Fitness
体育	运动	训练	健身	锻炼	体质

TABLE 10. Comparison of four algorithms on SogouCS long text dataset in F1 scores (%).

Number	TF-IDF	LAS	CNN	LCK-NFC (Ours)
1000	53.3	55.4	62.3	70.0
2000	55.8	57.1	65.9	73.6
3000	54.7	57.8	67.2	82.4
AVG	54.6	56.8	65.1	**75.3**

TABLE 11. Comparison of four algorithms on SogouCS short text dataset in F1 scores (%).

Number	TF-IDF	LAS	CNN	LCK-NFC (Ours)
10000	53.1	55.8	70.3	74.5
20000	51.5	56.2	74.7	82.8
30000	52.6	57.1	76.5	83.3
AVG	52.4	56.4	73.8	**80.2**

of total texts introduced. Experimental results show that the performance of our clustering algorithm suffers little with noise added. As the dataset increase, the degree of interference with clustering decreases. The reason is that the noise has less influence on the position of the cluster center when the dataset increase. Moreover, it is observed that the effect of clustering algorithm is not directly related to the amount of noise. Therefore, the proposed LCK-NFC algorithm has a certain degree of noise robustness.

F. COMPARISON AMONG DIFFERENT METHODS

To evaluate the performance of our proposal, we compare our proposed LCK-NFC algorithm with some popular feature extraction methods used in text clustering algorithms. Note that k-means algorithm is utilized as a clustering algorithm in the following comparisons.

Table 10 and Table 11 show the comparison between our proposed LCK-NFC algorithm and other text clustering methods, such as TF-IDF [6], LAS [7] and CNN [19] in F1 scores. The number of iteration and convolutional kernel is 100 in our experiments. The experimental results show that the proposed method performs best among all the methods. It should be noted that with texts increasing, the performance of LCK-NFC algorithm shows a notable upward trend, while the size of dataset has small impact on two traditional text clustering methods. Compared with CNN based clustering algorithm in [19], the proposed LCK-NFC algorithm achieves 15.7% and 8.7% improvement on long text and short text respectively. It is worth noting that LCK-NFC algorithm has a greater improvement on long text clustering, since the optimization in [19] is taken for short text. Therefore, our proposal is suitable for both long and short texts.

G. COMPARISON AMONG DIFFERENT CORPORA

Since ChnSentiCorp corpus consists of comment text without title, we only use the contents(or long texts) of every corpus as experimental dataset. Table 12 lists the F1 scores of LCK-NFC algorithm when processing different corpora with 100 iterations. The biggest difference between ChnSentiCorp corpus used in the experiment and the other two corpora is that both Fudan corpus and SogouCS have five categories, while ChnSentiCorp corpus has only two categories. It can be seen from the values in Table 12, with the same text numbers, the less the number of categories is, the better the clustering

TABLE 12. Comparison among different corpora on long texts in F1 scores (%).

Number	SogouCS	Fudan Corpus	Chnsenticorp
1000	70.0	72.8	81.6
2000	73.6	74.3	82.9
3000	82.4	81.5	84.3
AVG	75.3	76.2	82.9

performance can be. With the same text number, fewer text categories means more texts in each category, and therefore it is easier to find similarity between them. Moreover, when LCK-NFC algorithm deals with SogouCS corpus and Fudan corpus, the clustering performance shows the similar trend with the increase of text number.

H. EXAMPLES

In order to show vectorized representation and feature extraction of texts in LCK-NFC algorithm, Table 8 and Table 9 list the most important words extracted from different types of texts in SogouCS corpus and Fudan corpus. The most important words refer to the highest frequency words selected by the pooling layer. It can be concluded that, LCK-NFC algorithm is able to extract the most representative words to represent texts through continuous iteration. Thus, the clustering performance of LCK-NFC algorithm can be significantly improved compared with other methods.

V. CONCLUSION AND FUTURE WORK

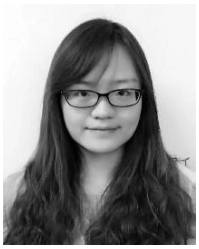
In this paper, we present a neural feedback text clustering algorithm based on BiLSTM-CNN-Kmeans architecture. By combining BiLSTM and CNN, text representations are generated with contextual semantics. A neural feedback clustering strategy is also proposed to dynamically adjust and optimize the network training. Experimental results demonstrate that our proposal can achieve satisfactory clustering results with a certain degree of noise robustness.

Although the proposed architecture achieves good performance on text clustering, It is still possible to make improvement. For example, k-means algorithm can be extended to other clustering algorithms to find a more general feedback model. In addition, long texts and short texts have different attributes, while we use the same algorithm. In the future, we would like to do more investigations and experiments to make up for these deficiencies.

REFERENCES

[1] C. C. Aggarwal and C. Zhai, "A survey of text clustering algorithms," in *Mining Text Data*. Boston, MA, USA: Springer, 2012, pp. 77–128.
[2] K. V. Kanimozhi and M. Venkatesan, "Survey on text clustering techniques," *Adv. Res. Elect. Electron. Eng.*, vol. 2, no. 12, pp. 55–58, 2015.
[3] P. G. Anick and S. Vaithyanathan, "Exploiting clustering and phrases for context-based information retrieval," in *Proc. ACM SIGIR Conf.*, 1997, pp. 314–323.
[4] S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis*, vol. 4. Amsterdam, The Netherlands: Springer, 1990.
[5] S. Seifzadeh, A. K. Farahat, M. S. Kamel, and F. Karray, "Short-text clustering using statistical semantics," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 805–810.
[6] J. L. Neto, A. D. Santos, C. A. A. Kaestner, and A. A. Freitas, "Document clustering and text summarization," in *Proc. 4th Int. Conf. Pract. Appl. Knowl. Discovery Data Mining (PADD)*, 2012, pp. 41–55.
[7] W. Song and S. C. Park, "Genetic algorithm for text clustering based on latent semantic indexing," *Comput. Math. Appl.*, vol. 57, nos. 11–12, pp. 1901–1907, 2009.
[8] S. Banerjee, K. Ramanathan, and A. Gupta, "Clustering short texts using wikipedia," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2007, pp. 787–788.
[9] X. Hu, N. Sun, C. Zhang, and T.-S. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 919–928.
[10] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. ACL*, 2015, pp. 1–11.
[11] N. Liu and J. Han, "A deep spatial contextual long-term recurrent convolutional network for saliency detection," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3264–3274, Jul. 2018.
[12] J. Han, G. Cheng, Z. Li, and D. Zhang, "A unified metric learning-based framework for co-saliency detection," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.
[13] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 84–100, Jan. 2018.
[14] G. Cheng, P. Zhou, and J. Han, "Duplex metric learning for image set classification," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 281–292, Jan. 2018.
[15] J. Han, R. Quan, D. Zhang, and F. Nie, "Robust object co-segmentation using background prior," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1639–1651, Apr. 2018.
[16] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.
[17] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published.
[18] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proc. COLING*, 2014, pp. 69–78.
[19] J. Xu *et al.*, "Short text clustering via convolutional neural networks," in *Proc. HLT-NAACL*, 2015, pp. 62–69.
[20] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, and J. Zhao, "Self-taught convolutional neural networks for short text clustering," *Neural Netw.*, vol. 88, pp. 22–31, Apr. 2017.
[21] M. B. Revanasiddappa, B. S. Harish, and S. V. A. KuMar, "Clustering text documents using kernel possibilistic C-means," in *Proc. Int. Conf. Cogn. Recognit.* Singapore: Springer, 2018, pp. 127–134.
[22] J. Tang, X. Wang, H. Gao, X. Hu, and H. Liu, "Enriching short text representation in microblog for clustering," *Frontiers Comput. Sci.*, vol. 6, no. 1, pp. 88–101, 2012.
[23] J. Yin and J. Wang, "A Dirichlet multinomial mixture model-based approach for short text clustering," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 233–242.
[24] S. V. Wazarkar and A. A. Manjrekar, "HFRECCA for clustering of text data from travel guide articles," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2014, pp. 1486–1489.
[25] W. Song, Y. Qiao, S. C. Park, and X. Qian, "A hybrid evolutionary computation approach with its application for optimizing text document clustering," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2517–2524, 2015.
[26] C. Jia, M. B. Carson, X. Wang, and J. Yu, "Concept decompositions for short text clustering by identifying word communities," *Pattern Recognit.*, vol. 76, pp. 691–703, Apr. 2018.
[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[28] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Empiricial Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
[29] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 2428–2437.

[30] C. Wu, F. Wu, S. Wu, Z. Yuan, J. Liu, and Y. Huang, "THU_NGN at SemEval-2018 task 2: Residual CNN-LSTM network with attention for English emoji prediction," in *Proc. The 12th Int. Workshop Semantic Eval.*, 2018, pp. 410–414.
[31] S. Song, H. Huang, and T. Ruan, "Abstractive text summarization using LSTM-CNN based deep learning," *Multimedia Tools Appl.*, vol. 10, pp. 1–19, 2018.
[32] P. C. Chang, M. Galley, and C. D. Manning, "Optimizing Chinese word segmentation for machine translation performance," in *Proc. 3rd Workshop Stat. Mach. Transl., Assoc. Comput. Linguistics*, 2008, pp. 224–232.
[33] C. Wang and B. Xu, "Convolutional neural network with word embeddings for Chinese word segmentation," in *Proc. 8th Int. Joint Conf. Natural Lang. Process. (IJCNLP)*, 2017, pp. 163–172.
[34] H.-P. Zhang, H.-K. Yu, D.-Y. Xiong, and Q. Liu, "HHMM-based Chinese lexical analyzer ICTCLAS," in *Proc. 2nd SIGHAN Workshop Chin. Language Process. Assoc. Comput. Linguistics*, vol. 17, 2003, pp. 184–187.
[35] C. J. van Rijsbergen, *Information Retrieval*. London, U.K.: Butterworths Scientific, 1975.
[36] C. J. Fox, "Lexical analysis and stoplists," in *Information Retrieval*. Upper Saddle River, NJ, USA: Prentice-Hall, 1992.
[37] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1995.
[38] Y. Kim. (Aug. 2014). "Convolutional neural networks for sentence classification." [Online]. Available: https://arxiv.org/abs/1408.5882
[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdi, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Nov. 2014.
[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[41] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, May 2013, pp. 6645–6649.
[42] M. Yao, D. Pi, and X. Cong, "Chinese text clustering algorithm based k-means," *Phys. Procedia*, vol. 33, pp. 301–307, Jan. 2012.
[43] R. Layton, *Learning Data Mining with Python*. Birmingham, U.K.: Packt, 2017.
[44] TSW. (2010). *Chnsenticorp Corpus*. Accessed: Feb. 8, 2015. [Online]. Available: http://www.searchofrum.org.co/tansongbo/coprus.html
[45] K. J. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier, "An exact algorithm for F-measure maximization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1404–1412.



YANG FAN received the B.E. degree in computer science from Shanghai Jiao Tong University, China, in 2016, and the M.E. degree in engineering from Waseda University, Japan, in 2017. She is currently pursuing the M.E. degree in information and communication engineering with Shanghai Jiao Tong University.

Her research interests include natural language processing and data mining.

LIU GONGSHEN received the Ph.D. degree in computer science from Shanghai Jiao Tong University in 2003.

After he joined the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, in 2004, he has been interested in natural language processing, data mining, and machine learning. In these research fields, he has published over 70 international articles in peer-reviewed journals and conferences. From 2014 to 2015, he was with Arizona State University, where he was involved in social network analysis with Professor S. Yau.

He is a Core Technical Personnel in information content analysis of the National Engineering Laboratory.

MENG KUI received the B.S. degree in automatic control from Shanghai Jiao Tong University, Shanghai, China, in 1995, and the Ph.D. degree in computer application from Fudan University, Shanghai, in 2006.

In 2006, she joined the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. She visited UCSD and ASU as a Visiting Scholar in 2010 and 2014, respectively. She has published over 40 articles. Her research interests include network and information system security assessment, and social networks.

SUN ZHAOYING received the M.S. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2018.

Her research interests include deep learning and natural language processing.