The Journal of Supercomputing (2023) 79:12266–12291
https://doi.org/10.1007/s11227-023-05139-w

CNN-VAE: An intelligent text representation algorithm

Saijuan Xu[1] · Canyang Guo[2] · Yuhan Zhu[2] · Genggeng Liu[2] · Neal Xiong[3]

Accepted: 23 February 2023 / Published online: 10 March 2023

Abstract

Collecting and analyzing data from all devices to improve the efficiency of business processes is an important task of Industrial Internet of Things (IIoT). In the age of data explosion, extensive text data generated by the IIoT have given birth to a variety of text representation methods. The task of text representation is to convert the natural language to a form that computer can understand with retaining the original semantics. However, these methods are difficult to effectively extract the semantic features among words and distinguish polysemy in natural language. Combining the advantages of convolutional neural network (CNN) and variational autoencoder (VAE), this paper proposes an intelligent CNN-VAE text representation algorithm as an advanced learning method for social big data within next-generation IIoT, which help users identify the information collected by sensors and perform further processing. This method employs the convolution layer to capture the local features of the context and uses the variational technique to reconstruct feature space to make it conform to the normal distribution. In addition, the improved word2vec model based on topical word embedding (TWE) is utilized to add topical information to word vectors to distinguish polysemy. This paper takes the social big data as an example to illustrate the way of the proposed algorithm applied in the next-generation IIoT and utilizes Cnews dataset to verify the performance of proposed method with four evaluating metrics (i.e., recall, accuracy, precision, and F1-score). Experimental results indicate that the proposed method outperforms word2vec-avg and CNN-AE in K-nearest neighbor (KNN), random forest (RF), and support vector machine (SVM) classifiers and distinguishes polysemy effectively.

Keywords Convolutional neural network · Feature extraction · Text representation · Topical word embedding · Variational autoencoder

✉ Genggeng Liu
liugenggeng@fzu.edu.cn

Extended author information available on the last page of the article

1 Introduction

Recently, the rapid development of Industrial Internet of Things (IIoT) results in the generation of massive data every day, including text, image, video, audio, etc. [1–3]. Text data play a significant role, because it not only occupies a large part of the Internet data but also can be applied in many real-world scenarios, including getting the current hot spots, developing question answering system and machine translation. Taking the search engine as an example, there are tens of millions of search task requests every day, most of which use text information as the input of search task [4]. The emergence of miniature low-cost sensors and high-bandwidth wireless networks in the IIoT era means that even the smallest devices can be connected today by digital intelligence with a certain level [5]. Then monitor and track them, share their status data, and communicate with other devices. Finally, all these data can be collected and analyzed to improve the efficiency of business processes. In this process, the analysis of text data is a very important part.

As the basic task of natural language processing (NLP), text representation transforms unstructured natural language into a structured form that contains unique semantic information of the original text data and thus can be processed and analyzed by computer [6]. According to the granularity of natural language, text representation can be categorized as four different granularity representations, word representation, sentence representation, paragraph representation, and document representation. For different granularity of text representation, they have the same purpose of extracting the most important semantic information in different applications [7–9]. In traditional text representation method, the bag-of-words model that takes each word in the dictionary as a feature of text representation has been widely used [10, 11]. A document is composed of text feature vectors corresponding to all words in the dictionary. Considering that the bag-of-words model uses all words, the dimension of the final text feature vectors will lead to dimension disaster with the scale of data increasing, which will take up a lot of computational costs and running time [12]. Therefore, a reasonable text representation method is necessary for NLP and feature extraction method is a useful approach to decrease the dimension of text feature vectors [13].

One-hot coding is a common method of text representation, which uses an n-dimensional vector to represent each word, and n is the number of non-repeated words in the corpus. The position of the vector corresponding to the word is set to 1, and the remaining positions are set to 0. Although this method can express text simply and clearly, it not only causes dimension disaster, but also is difficult to capture text information between contexts. For instance, assume that the word vectors of “pleased” and “happy” are coding by one-hot as [1,0,0,0] and [0,1,0,0]. “pleased” and “happy” have similar meanings, but the cosine similarity of the two word vectors is 0. Obviously, the similarity of the words calculated by one-hot is unreasonable, and their semantic information cannot be reflected. Besides, polysemy in natural language is also worthy of attention. Polysemy means that the same word can express multiple meanings, which is the general ambiguity in natural language. For example, “apple” refers to a technology company in

scientific articles and a fruit in nutrition magazines. Obviously, when words and word vectors are corresponding one by one, the computer cannot recognize the true meaning of the word.

Text feature extraction, as a learning algorithm of data preprocessing, can not only effectively reduce the dimension of feature space, but also directly affect the accuracy of subsequent classification algorithms [14]. Besides, polysemy in natural language is also worthy of attention. To effectively extract the semantic features among words and distinguish polysemy in natural language, this paper proposes an intelligent text representation algorithm by combining the advantages of convolutional neural network (CNN) and variational autoencoder (VAE). The proposed method has a lower computational cost, and it can extract semantic information and distinguish polysemy. The contributions of the paper are as follows.

- To effectively extract the semantic features among words, this paper proposes an intelligent CNN-VAE text representation algorithm, which extracts text features efficiently for text classification. This method employs the convolution layer to capture the local features of the context and uses the variational technique to reconstruct feature space to make it conform to the normal distribution.
- To effectively distinguish polysemy, this paper utilizes the improved word2vec model based on topical word embedding (TWE) to add topical information to word vectors. This paper equips a word with multiple topic vectors to match different semantics in different contexts. A word is mapped into multiple vectors corresponding to different topics by this model to solve polysemy. Accurate recognition of polysemy improves the text semantic extraction capability of CNN-VAE.
- This paper proposes a text representation application system for social big data within next-generation IIoT, which helps users to identify the information collected by sensors and perform further processing (e.g., news classification, sentiment analysis, public opinion analysis, e-mail filtering, etc.).
- This paper adopts recall, accuracy, precision, and F1-score as evaluation metrics and utilizes classification to evaluate the performance of the proposed method. And this paper uses K-nearest neighbor (KNN), random forest (RF), and support vector machine (SVM) as classifiers. Finally, the experimental results demonstrate that the proposed method is superior to comparative models and distinguishes polysemy effectively.

The rest of this paper is structured as follows. Section 2 represents the related work. An intelligent CNN-VAE text representation algorithm is proposed in Sect. 3. To distinguish polysemy, Sect. 4 further proposes a CNN-VAE text representation algorithm based on TWE. Section 5 illustrates the experimental results and analysis. The conclusions and future research directions are elaborated in Sect. 6.

2 Related work

The development of computer processing power promotes the application of neural networks in NLP to a certain extent. The neural network model based on deep learning is computationally complex and time-consuming. By increasing the computational depth of the neural network model, deep learning can use the computing power of the current machine to make the model better fit the desired result. Various researches showed that NLP could achieve good performance in neural networks, and better effect could be achieved by using neural networks for text representation. There are many application tasks and scenarios in NLP based on neural networks which have better performance in language model, machine translation, text classification, question answering system, word distributed representation, sentiment analysis, and topic marking [15, 16], and a sequence of theoretical research gains has been achieved. Among them, text representation as the basic task of these applications is concerned by the public.

2.1 Neural networks for text representation

Common applications of neural networks used in NLP are predicting the next word by context to develop models from the perspective of probability. The researchers proposed a word embedding language model based on neural networks (i.e., distributed representation) to settle the problem of word vectors [17]. The model was designed to predict experimental words and the word vectors were just the by-product of the optimized model. After that, adopting neural networks to capture text features became a research hot spot. Mikolov et al. [18] developed a word vector training model, namely word2vec. As the term suggests, the objective of word2vec was changing words to vectors. According to different granularity, there are four categories of text representation methods based on neural networks (i.e., words, sentences, paragraphs, and documents). Considering the data sparseness, dimension disaster, and lack of semantic expression in the bag-of-words model used in sentences and documents, multilayer neural networks provide an effective method to map and capture features. Many researchers have an interest in using word embedding, especially word2vec as the features of text classification tasks. However, since the word2vec features are high dimensional, it would lead to an increase in the complexity of the classifier. Therefore, Alshari et al. [19] put forward an effective method of feature extraction based on word2vec for sentiment analysis. Since the feature vector of each text was constructed from the polarity clusters, a lower-dimensional vector could be utilized to represent the text. Ji et al. [20] developed a word vector training model, namely Wordrank. It was good at representing similar words and could compute the similarity of words to acquire word vector representation. For learning the unlabeled text data, Hill et al. [21] developed a text representation method based on sentence, which improved the portability and reduced the amount of calculation. Since CNN was an expert in capturing local features [22, 23], Hu et al. [24] employed CNN to

obtain semantics among context for sentence matching tasks. Since the gathered tweets and news articles contain data that is not needed for learning, Jang et al. [25] employed two word embedding methods, including Skip-gram and continuous bag-of-words (CBOW), and established a Skip-gram-based CNN model and a CBOW-based CNN model. They evaluated the model classification accuracy of CNN without word2vec, CNN with Skip-gram, and CNN with CBOW in tweets and news articles. The experimental results showed that the word2vec was significant to classification model. Compared with Skip-gram, CBOW had better accuracy, while Skip-gram was smart in tweets. CNN is an artificial neural network with weight sharing, whose convolution layer can effectively extract local features. At present, CNN has been applied to many fields, including NLP, computer vision, etc. [23, 26–29]. Considering the multi-granularity phenomenon in natural language, Yin et al. [30] proposed a text representation model, namely Bi-CNN-MI. The proposed method could not only extract text representation with different granularity but also realize synonymous sentence detection. Since long short-term memory (LSTM) can handle time-series data well, Luan et al. [31] developed a text classification model based on CNN and LSTM. They collected the subjective and objective text categorization dataset to verify the performance of CNN and LSTM, and the experimental results showed the superiority of the proposed method. Although a particular model can capture more exact features, contextual information is ignored. Therefore, Shi et al. [32] provided a solution by developing a C-LSTM with word embedding model. A large number of experimental results showed that this method had excellent performance in Chinese news text classification. On the basis of LSTM, Bai [33] developed the LSTM text classification method with an attention mechanism. This method utilized the input gates, forget gates, and output gates of LSTM to extract long-term dependence of context. Attention mechanism was added to obtain more contextual semantic information, and the results showed that this method could improve the prediction accurateness effectively. Li et al. [34] provided a deep network, namely Bi-LSTM-CNN, which employed recursive structure to gain the contextual information and used CNN to construct the context of per word. In NLP, CNN uses a convolution kernel to convolve text matrices of various lengths and uses a pooling layer to calculate the vectors passing through convolution kernel to capture features for text classification [35]. Ameur et al. [36] utilized the ability of dimension reduction and feature extraction of autoencoder (AE) to predict emotional polarity tags at word and sentence levels, which enhanced the accuracy of emotion analysis tasks greatly. To enhance the accuracy of sentiment classification, Lu et al. [37] developed a VAE-based method. AE is an unsupervised feature learning way, which aims at extracting features from raw data [36]. The variational technique was utilized to reconstruct feature space to make it conform to the normal distribution. VAE is a deep generative model for unsupervised and semi-supervised learning tasks because it is an expert in processing unlabeled data in learning tasks [38]. Yu et al. [39] proposed a one-dimensional residual CNN-AE, which had better performance compared with deep belief network, stacked AE and CNN when capturing features from vibration signals under unsupervised learning.

2.2 Topic model for polysemy

Considering the polysemy of a word in natural language, Hoffman et al. [40] presented the latent Dirichlet allocation (LDA) model, a topic model, which could extract the topic information from an article. Considering the explosive growth of text data, Guven et al. [41] developed a two-stage method according to the classical LDA algorithm. To measure the usefulness of machine learning ways in Weka, they developed a file with an arff extension in accordance with the word weight of topic. Considering that it is a heavy task to use a large amount of data generated by the Internet to analyze and predict the market, Kanungsukkasem et al. [42] developed a LDA-based topic model which aimed to capture features from news, articles, etc. Experimental results demonstrated that LDA-based topic model was superior to contrast methods. Sohail et al. [43] developed an index to verify the sensitivity of monetary policy communication. The index identified the positive or negative value of a sentence and applied LDA algorithms to capture monetary policy data and recognize words. DiMaggio et al. [44] used LDA to analyze how to elaborate a policy area (i.e., government assistance to art organizations and artists) in nearly 8,000 articles. They explained the advantages of topic model as a method of processing large text corpus and emphasized the connection between topic modeling method and core concepts in cultural studies, such as cultural framework, ambiguity, heterogeneity, and meaning relations. Hsu et al. [45] presented a hybrid LDA method to improve the topic classification performance and genetic algorithm (GA) was adopted to optimize weights. Novichkova et al. [46] proposed an NLP engine for general biomedical field, namely MedScan. The engine employed a specifically exploited context-free grammar and dictionary, which could effectively handle sentences in the Medline abstracts and generate a series of standardized logical structures to express the sense of per sentence. An initial evaluation of accuracy, system performance, and coverage showed excellent results. They also researched other methods to reduce ambiguity resolution and enhance coverage. Wu et al. [47] proposed a frontier knowledge discovery model based on LDA and knowledge organization system (KOS), which could transform biomedical literature into structured data for detecting emergent topics and their semantic information relationships in the field of cancer. Experimental results proved that the combination of LDA and KOS could effectively eliminate the noise of the semantic layer and achieve good results in visualization, evolutionary recognition, and topic recognition. On the basis of Skip-gram model, Liu et al. [48] developed a TWE-based model. This model introduced topical vectors while training word vectors, which aimed to obtain different word vector representations under different topics by topical vectors. Bordes et al. [49] proposed a neural network that could embed multiple-relational graphs into a flexible successive vector space to preserve and enhance the raw data. The training network encoded the semantics of these graphs and assigned them to reasonable components with high probability. This method could eliminate word ambiguity in the context of text semantic analysis. The experimental results proved that the method had good performance on literature dataset and real-world knowledge base.

3 Intelligent CNN-VAE text representation algorithm

In the existing research, neural networks have achieved excellent results in the processing of audio and images. However, there are still many unresolved problems or areas that need to be optimized in the application of deep learning due to the ambiguity of text and the complexity of natural language in NLP. Based on the above research and related analysis of NLP, this paper focuses on the representation of text features under the neural network method and learns the feature extraction in the text. Then, this paper extracts the information features in the text from the document vector and the word vector, respectively, and applies them to text classification.

Due to the high dimensional and high sparse of the document vector generated by the bag-of-words model, this paper extracts low-dimensional and dense features from the text vector through AE. The text feature representation can extract important information of the original text, which can effectively improve the efficiency of text processing. In addition, this paper introduces an improved model of the AE, namely the VAE model, which uses the prior assumption of normal distribution to make the vector distribution of the text feature representation space more in line with the normal distribution and further improves the semantic information represented by the final text features [50]. Furthermore, this paper combines the network framework of CNN and VAE to use an unsupervised method to extract the text feature representation from the word vector, which makes the final text feature representation has more semantic information.

The CNN can capture local features, so it is often used for text data learning in text feature representation. The VAE model is proposed as a generative model, which assumes that the variables of the hidden layer obey a normal distribution, and the subsequent generated data are all sampled from this distribution. Since VAE is an extension of the AE model, it has the same characteristics as the AE model; that is, its hidden layer can be extracted as a feature of the data. In this section, VAE is used to extract features of text data; that is, the text variables are input into the VAE model for training. By constraining the number of neurons in the hidden layer, the original high-dimensional and sparse text vector is transformed into a low-dimensional and dense text vector, and then, text classification is performed in the newly generated text vector space. How to combine the CNN structure and the VAE mechanism to construct a hybrid model with the advantages of the two models, so as to optimize the learning process of the network, is the research goal of this section.

The structure of VAE can be divided into input layer, hidden layer, and output layer. VAE restricts the hidden layer, as shown in Fig. 1, and it assumes that the hidden layer obeys normal distribution. All hidden variables z are sampled from this normal distribution, and the raw data x are reconstructed. VAE hopes to make the generated data close to raw data by optimizing the parameters θ to maximize $p_\theta(x)$. This process can be expressed as follows.

$$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz, \tag{1}$$

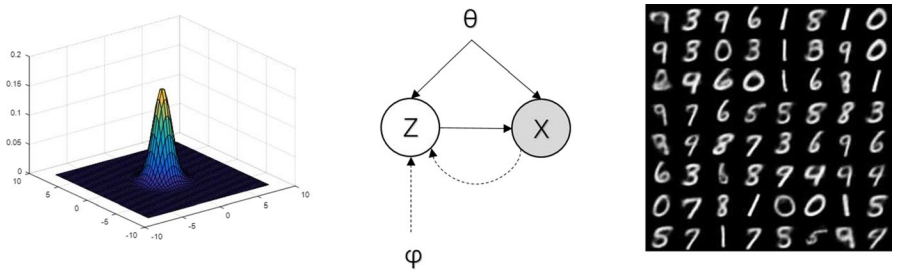


Fig. 1 Probabilistic graphical model of VAE

$$D_{\mathrm{KL}}(q_\varphi(z|x) \parallel p_\theta(z|x)) = \log p_\theta(x) + \sum q_\varphi(z|x)\left[\log q_\varphi(z|x) - \log p_\theta(z) - \log p_\theta(x|z)\right], \tag{2}$$

$$-D_{KL}(q_\varphi(z|x \parallel p_\theta(z)) = \frac{1}{2}\sum(\log(\sigma^2 - \mu^2 - \sigma^2 + 1)), \tag{3}$$

$$\sum q_\varphi(z|x)[\log p_\theta(x|z)] = \log p_\theta(x|z), \tag{4}$$

$$z = \sigma \cdot \varepsilon + \mu, \tag{5}$$

$$\log p_\theta(x|z) = -\sum\left(\log\left(\sigma\sqrt{\pi x}\right) + \left(\frac{1}{2}\left|\frac{x-\mu}{\sigma}\right|\right)\right), \tag{6}$$

$$\begin{aligned} &L(\theta, \varphi; x) \\ &= -\sum q_\varphi(z|x)\left[\log q_\varphi(z|x) - \log p_\theta(z) - \log p_\theta(x|z)\right] \\ &= \sum q_\varphi(z|x)[\log p_\theta(x|z)] - D_{KL}(q_\varphi(z|x) \parallel p_\theta(z|x)) \\ &= \frac{1}{2}\sum(\log(\sigma^2 - \mu^2 - \sigma^2) + 1) \\ &\quad - \sum\left(\log\left(\sigma\sqrt{\pi x}\right) + \left(\frac{1}{2}\left|\frac{x-\mu}{\sigma}\right|\right)\right), \end{aligned} \tag{7}$$

where $p_\theta(x|z)$ is a set of functions which aims to generate x from z. θ is the parameter of the model. x is a matrix which is composed of the word vectors of the words appearing in the article. z is a random number which is generated by the mean value μ and variance σ. Kullback–Leibler (KL) divergence is introduced to evaluate the similarity of different distributions [51]. $p_\theta(z|x)$ is the encoder which is utilized to get $p_\theta(z)$. Considering $p_\theta(z|x)$ is hard to calculate, this paper adopts an approximate posterior $q_\varphi(z|x)$ instead of $p_\theta(z|x)$. $\sum q_\varphi(z|x)[\log p_\theta(x|z)]$ means reconstruction error. $-D_{KL}(q_\varphi(z|x) \parallel p_\theta(z|x))$ denotes regularizer. This paper utilizes the Monte Carlo evaluation to settle the reconstruction error. Since z is not derivable, this paper

adopts the reparameterization technology which introduces the derivation of μ and σ to take the place of the original derivation of z. ϵ can be considered as sampling from $N(0, 1)$. $L(\theta, \varphi; x)$ is the cost function, and the goal of the CNN-VAE model is to minimize this loss function.

Considering the advantage of CNN in extracting local features, the fully connected mode of AE is replaced by convolutional connection. Combining the advantages of CNN and VAE, this paper proposes an intelligent CNN-VAE text representation technology (shown in Fig. 2) for social big data in semantic networks, which helps users to identify the received information and do further processing. The proposed algorithm employs the convolution layer to capture the local features of the context and uses the variational technique to reconstruct feature space to make it conform to the normal distribution. The computational process of CNN-VAE is similar to AE. The input vectors are decoded by convolution layer, pooling layer, and fully connected layer. Algorithm 1 shows the training process of using the CNN-VAE model for feature extraction of text data.

Algorithm 1 The training process of the CNN-VAE model.

Require: The matrix $[x_i]_{i=1}^n$ formed by the word vectors of the words in the article, the original dimension n, the output required dimension d

Ensure: The reconstructed matrix $[\hat{x}_i]_{i=1}^n$

1: Set the size of the convolution kernel, the number of convolution kernels and the step size.
2: Extract local features (i.e. the word vector matrix corresponding to the convolution kernel) through the convolution layer to obtain the feature vectors of the matrix.
3: Extract the feature value with the highest semantic information among the local features in the window by max pooling.
4: Use the activation function to predict the corresponding label of the word vector.
5: After convolution, pooling, and fully connected layers, the feature extraction of the encoder is obtained.
6: Generate a normal distribution by reparameterization and sample it to obtain the feature $[z_i]_{i=1}^d$ output by the encoder.
7: The text feature $[z_i]_{i=1}^d$ after feature extraction is deconvolved to obtain the reconstructed matrix $[x_i]_{i=1}^n$.
8: **return** $[z_i]_{i=1}^d$

4 CNN-VAE text representation algorithm based on TWE

Polysemy of a word is widespread in natural languages, especially in Chinese. A word can express multiple meanings, which is the ambiguity in natural language. The task of semantic disambiguation is also an important research direction in NLP.

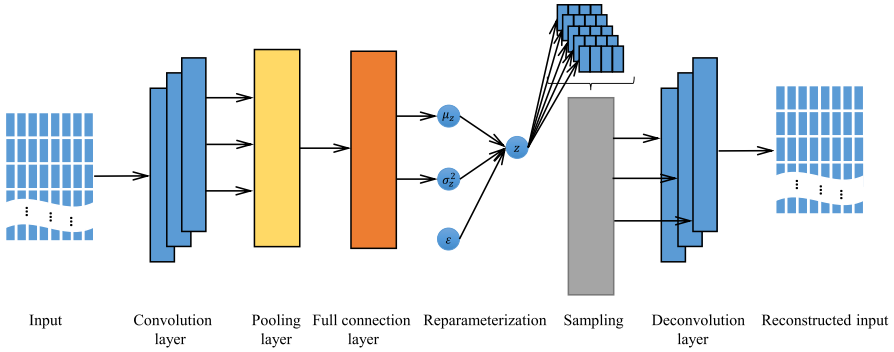


Fig. 2 The structure of CNN-VAE model

A word vector generated by word2vec corresponds to only one word in the corpus, which ignores the ambiguity in natural language [52]. Therefore, an improved word-2vec model based on TWE is proposed. A word is mapped into multiple vectors responding to different topics by this model according to Bayesian probability to solve polysemy. In this model, word vectors generated by word2vec on the topic model are used as the input of CNN-VAE model. On this basis, this paper establishes a CNN-VAE text representation algorithm based on TWE.

4.1 Word2vec

The basic idea of word2vec is to represent each word as a real number vector. It gains the weight matrix of the network via training a neural network, thereby gaining the text word vector model. Word2vec uses CBOW (Fig. 3) and Skip-gram (Fig. 4) to train the model and gain word vectors. The training process mainly has three stages: input, projection, and output. Skip-gram forecasts the probability of the context via the specific word, while CBOW forecasts the probability of a specific word via the context of the specific word. The word vectors generated by word2vec cannot solve the problem of polysemy in natural language. In word2vec, a word has only one word vector, but in fact, some words in natural language contain many different meanings and even some words in different contexts represent far different semantic information. Therefore, this paper introduces the improved word2vec model based on TWE, which can distinguish the different semantics of the word vector under different topics. The text representation obtained by this model can solve the polysemy of word vector input and increase the speed of calculation.

4.2 LDA

The hypothetical structure of LDA is that different words make up a topic and different topics make up a document, namely "document-topic-word" structure. LDA is a three-layer Bayesian probability model based on this structure [40].

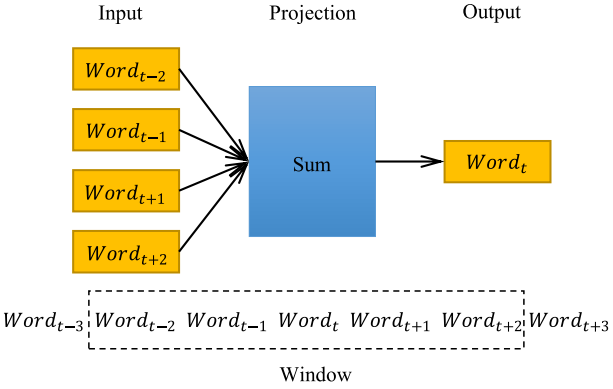


Fig. 3 The structure of CBOW model

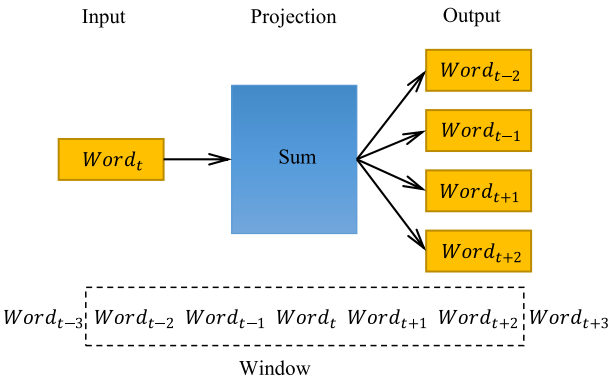


Fig. 4 The structure of Skip-gram model

Figure 5 shows the structure of LDA model. Assuming that there are k topics in a document set D, and each document is made up of these k topics according to different probabilities, the matrix storing the corresponding probabilities is the document-topic matrix. Similarly, each topic is also made up of m words according to different probabilities, and the matrix storing the corresponding probabilities is the topic-word matrix.

As shown in Fig. 6, two stages make up the Bayesian probability model. In the first stage, the topic distribution θ_{d} of document d is generated by sampling from the Dirichlet distribution α, which can be expressed as $\theta_{\mathrm{d}} \sim Dir(\alpha)$. The topic polynomial distribution generates the topic Z of each word in document d. In the second stage, the LDA topic model samples from the Dirichlet distribution β to generate the word polynomial distribution φ_Z of topic Z which is expressed as $\theta_Z \sim Dir(\beta)$. The word polynomial distribution φ_Z generates the final word W. α and β are the prior parameters of Dirichlet distribution of document-topic and topic-word. θ_{d} represents the topic distribution in document d, and Z represents

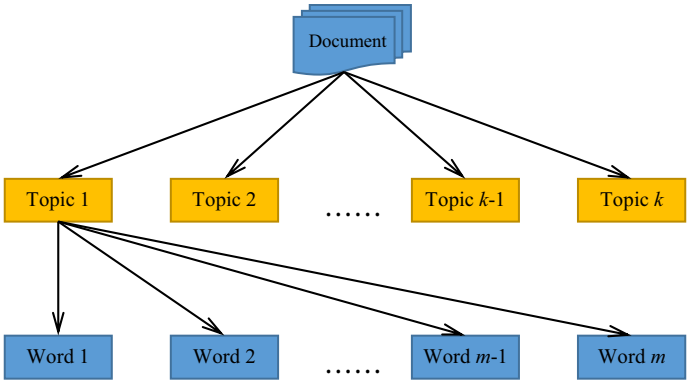


Fig. 5 The structure of LDA model

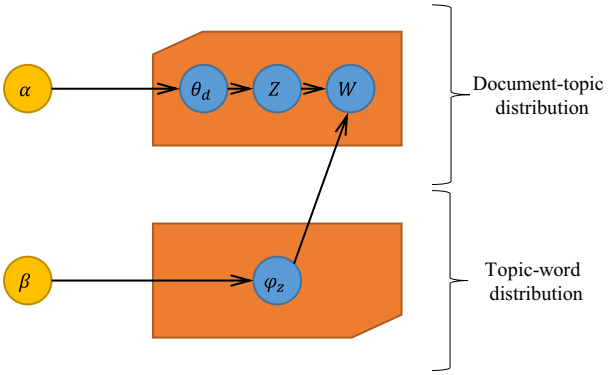


Fig. 6 Parameter relationship of LDA model

the corresponding topic set. φ_Z represents the word components contained in topic Z. According to the principle of LDA topic, a document can be obtained from the probability distribution of document-topic. That is to say, for document $D(i, j)$ (i.e., $j-th$ topic in topic set i) can be obtained by polynomial distribution $D(i,j) \sim Mult(\theta_d)$. Topic $Z(i, j)$ corresponding to the $j-th$ word in document i can be obtained through the polynomial distribution $Z(d, i) \sim Mult(\varphi_Z)$.

In the parameter setting of LDA topic model, the prior parameters α and β of Dirichlet distribution are set according to experience. The posteriori parameters θ_{d} and φ_Z of the polynomial distribution need to be estimated by calculating the corresponding posteriori probability distribution from the data in the known corpus. Therefore, this paper employs Gibbs sampling to calculate the posteriori parameters θ_{d} and φ_Z. Considering the real data is usually difficult to find out the corresponding accurate probability distribution. Therefore, the approximate inference method is often used to randomly fit the real probability distribution by sampling. Gibbs sampling is based on above idea, and it samplings m *n-dimensional* data $[X_i]_{i=1}^n$ from a joint probability distribution [53]. The vectors

X_i obtained by sampling are initialized randomly. Each sample X_i can be derived from the conditional probability distribution $P(X_i^j|X_i^j, ..., X_i^j, X_{i-1}^{j+1}, ..., X_{i-1}^n)$, and X_i^j denotes the value of the *j-th* dimension of sample X_i. Sampling formula of Gibbs sampling can be shown as

$$P(Z_i = K|Z_{i-1}, W) \propto \frac{\left(n_{k-i}^{(i)} + \beta_i\right)\left(n_{d-i}^{(k)} + \alpha_k\right)}{\left(\sum_{i-1}^{V} n_{k-i}^{(i)} + \beta_i\right)}. \tag{8}$$

After substituting the parameters α and β of LDA topic model, the posterior probability distribution of LDA topic and word can be computed as

$$P(z, w|\alpha, \beta) = \prod_{z=1}^{T} \frac{\Delta\left(n_z + \beta\right)}{\Delta\beta} * \prod_{d=1}^{D} \frac{\Delta\left(n_{\mathrm{d}} + \alpha\right)}{\Delta\alpha}. \tag{9}$$

When Gibbs sampling algorithm converges, the document-topic probability distribution θ_d and topic-word probability distribution φ_Z can be computed as follows.

$$\theta_{d,z} = \frac{n_{\mathrm{d}}^{Z} + \alpha}{\sum_i^V n_z^i + \beta}, \tag{10}$$

$$\varphi_{z,i} = \frac{n_z^i + \beta}{\sum_{i=1}^{V} n_z^i + \beta}. \tag{11}$$

Finally, the topical probability distribution of a document and the word probability distribution of per topic would be obtained, so as to realize the topical mining of the document.

4.3 Topical word model

Word representation is the minimum granularity representation in text representation, and word vector representation model word2vec is applied in various fields of NLP. However, the word vectors generated by word2vec cannot solve the problem of polysemy in natural language. In word2vec, a word has only one word vector, but in fact, some words in natural language contain many different meanings and even some words in different contexts represent far different semantic information. Therefore, the word vector representation of word2vec is a little unreasonable. On the basis of Skip-gram model, the model based on TWE introduces topical vectors while training word vectors, which aims to achieve different word vector representations under different topics through the topical vectors [48]. Each topic in the model is trained as a word, and the model learns the topical embedding of topic z_i and word embedding of word w_i separately. Then, the topical word embedding $\langle w_i, z_i \rangle$ is trained according to topical embedding of

z_i and word embedding of w_i. TWE aims to study the vector representations of words and topics at the same time, and its structure is shown in Fig. 7.

Compared with Skip-gram using the central word w_i in the sliding window to predict context, this model employs central word w_i adding semantic information (i.e., topic z_i) to predict context. This model aims to solve the polysemy by changing the words into their corresponding vectors in different topics. The topical word embedding of word w in topic z can be obtained by connecting word embedding with topical embedding as expressed as

$$w^z = w \oplus z, \tag{12}$$

where $\oplus$ is a cascading operation and the vector dimension of w^z is twice that of w or z.

5 CNN-VAE text representation system based on TWE

The input of CNN-VAE model is the word vector pretrained by word2vec, which cannot solve polysemy. This paper equips a word with multiple topic vectors and matches different semantics in different contexts. Then, the features of these word vectors are extracted through CNN-VAE. Therefore, an improved word2vec model based on TWE is proposed to get the word vectors meeting the requirements, as shown in Fig. 8. The proposed model uses LDA to train each word in the text to get its corresponding topical number. The topical vectors and the word vectors are trained depending on the TWE model. Lastly, the input vectors of CNN-VAE are generated. Then, we use the extracted features for classification, thereby verifying the ability of text feature extraction and the effectiveness of the CNN-VAE text representation algorithm based on TWE. Algorithm 2 shows the process of generating topic-word vectors using the word2vec model based on TWE in this paper.

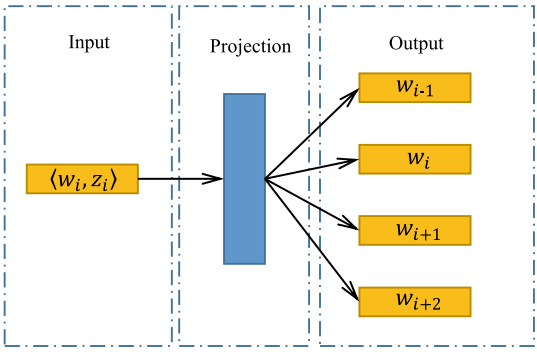


Fig. 7 The structure of TWE model

Algorithm 2 The training process of the improved TWE-based word2vec model.

Require: The word $word_i, i = \{1, 2, ..., n\}$ in the text
Ensure: The topic-word vector, w^z

1: Train $word_i$ through the LDA model to get its corresponding topic number.
2: Convert words in the text to $\langle word_i : topic \rangle$.
3: Train the text to get the topic vector z and word vector w through the TWE model.
4: Generate CNN-VAE input vector w^z according to $\langle word_i : topic \rangle$ and Equation (12).
5: **return** w^z

Based on above technology, this paper proposes a CNN-VAE text representation system based on TWE (shown in Fig. 9), which is an advanced learning method for social big data learning within next-generation IIoT. The architecture of text representation application system is shown in Fig. 10. First, text data produced by media and individuals are sensed and stored by sensors in electronic devices. Then, these text data are sent to users via PC or mobile phone. Next, the users send inquiries about text processing to the cloud. The text processing server extracts text features (i.e., contributions of this paper) for text processing (e.g., news classification, sentiment analysis, public opinion analysis, e-mail filtering, etc.) and replies to users with relevant information.

6 Experimental results and analysis

6.1 Dataset and preprocessing

The data preprocessing is shown in Fig. 11. The label message (i.e., news category) is fetched from the dataset, and the dataset is split into label and corpus. Next, in

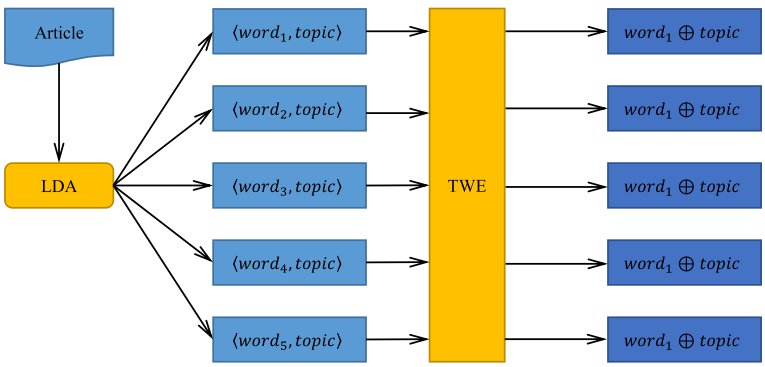


Fig. 8 Improved word2vec model based on TWE

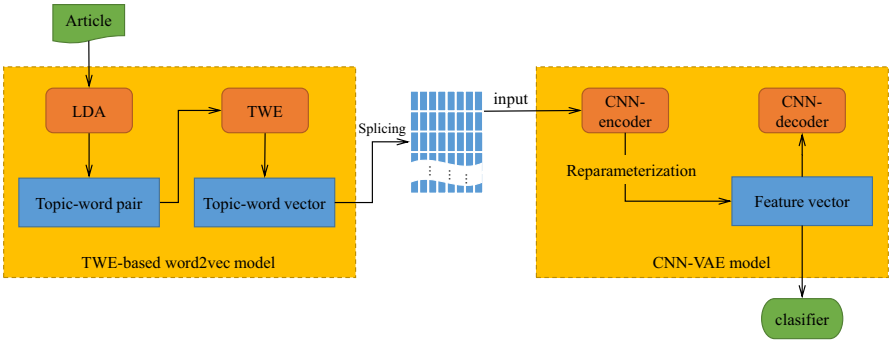


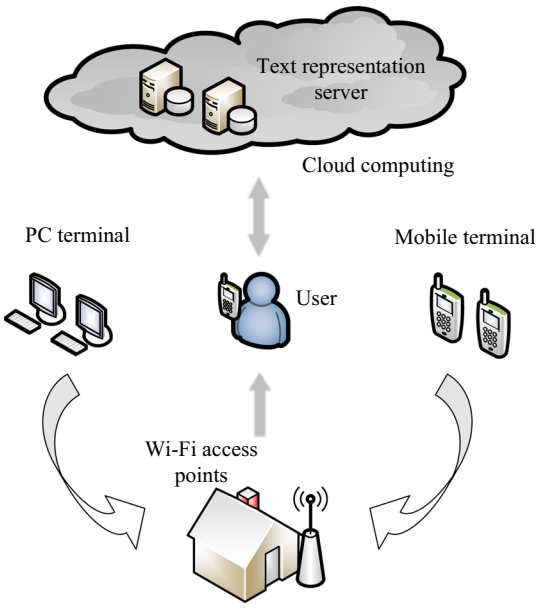
Fig. 9 CNN-VAE text representation algorithm based on TWE

order to raise the performance and calculated efficiency of following algorithms, the article is handled by using Jieba segmentation [54], and a number of words are filtered out by using the stop words. In addition, in this paper, the word frequency statistics of the words in the corpus are carried out, and the word frequency is sorted in descending order. Ultimately, the words, whose word frequency is lower than the first 10000 words, are removed to gain the last dataset.

6.2 Pretraining

The network training process consists of three parts. First, when minimizing the loss function, use the training set data to seek the best parameters of the model. Next, put

Fig. 10 The architecture of text representation application system



the verification set into the encoder network obtained from the training set to obtain the corresponding text representation vector when the network loss converges. Then, input the text representation vector into the classification model to gain the corresponding classification accuracy. Finally, the best parameters of the model are got by setting different learning rates constantly. The best model parameters are obtained after several adjustments. The dimension of word vector is 128, the number of iterations is 30, the dropout is 0.5, the size of hidden layer is 128, the learning rate is 0.001, and the padding (i.e., the maximum number of words in each article) is 100. On the basis of the obtained best parameters and structure, the test set data are textually expressed and then input into the classification algorithm, and multiple evaluation indicators are used to verify the performance of the network.

Before using LDA topic model to generate topics, the paper needs to determine the number of topics. In this paper, two methods are used to determine the number of topics.

- Calculate perplexity of different number of topics.
- The probability of each document under all topics is calculated and transformed into its corresponding document vector, which is used to input into SVM classifier to compare the classification accuracy in different topics.

The final number of topics is determined by comparing the above two methods. The degree of perplexity is inversely proportional to the fitting ability of the model, the smaller the degree of perplexity of the model, the better the fitting effect on the text. In the LDA topic model, its perplexity can be calculated as

$$\text{perplexity}(D) = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(w_{\text{d}})}{\sum_{d=1}^{M} N_{\text{d}}}\right\}, \tag{13}$$

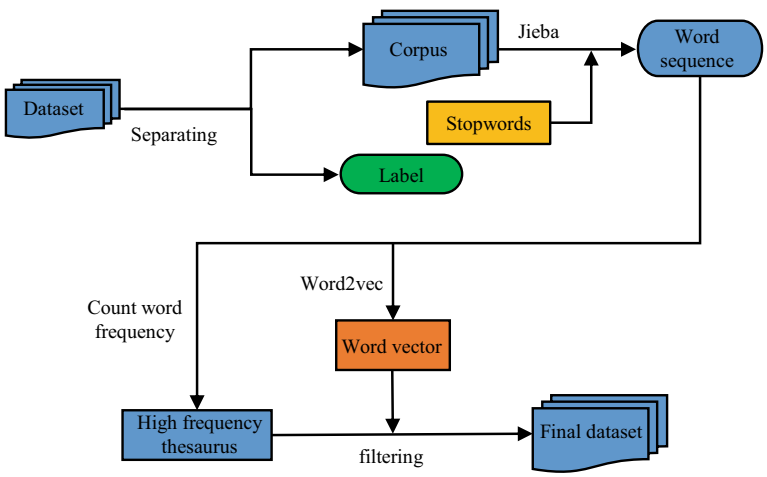


Fig. 11 Data preprocessing

where D is the test set in the corpus, N_{d} is the number of words in each document, w_d is the word in document d, and $P(w_d)$ is the probability of generating words w_{d} in the document.

6.3 Evaluating metrics

To verify the performance of the proposed model, the paper uses the following evaluation metrics: recall, accuracy, precision, and F1-score, and their values are defined as follows:

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{14}$$

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}, \tag{15}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{16}$$

$$F1 - \mathrm{score} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}, \tag{17}$$

where the true positive (TP) means that the predicted category and the actual category are both positive. The false positive (FP) means that the predicted category is positive, while the actual category is negative. The false negative (FN) means that the predicted category is negative, while the actual category is positive. The true negative (TN) means that both the predicted category and the actual category are negative.

6.4 IIoT applications

The paper uses the open dataset Cnews [55], which is obtained by screening the historical data of Sina News RSS subscription channel from 2005 to 2011. There are 10 types of news in the dataset, namely current affairs, education, entertainment, estate, fashion, finance and economics, games, home furnishing, science and technology, and sports. There are 65000 text data in the dataset, which are split into 50000 training data, 10000 testing data, and 5000 validation data as shown in Table 1. In the dataset, each line corresponds to an article, and the start of each line represents the news category of that article.

The paper uses the training set to train the LDA model and selects 1% of them randomly as the validation set to calculate the degree of perplexity. The number of topics is set from 1 to 80. The relationship between the degree of perplexity and the number of topics is shown in Fig. 12, and the relationship between the classification accuracy and the number of topics is shown in Fig. 13. In Fig. 12, the degree

of perplexity attains the minimum when the number of topics is 65, which indicates that when the number of topics is set to 65, the model is the best for data fitting and prediction. In Fig. 13, when the number of topics is less than 45, the number of topics is proportional to accuracy. When the number of topics is higher than 45, the accuracy fluctuates in a certain range. Therefore, it can be concluded that the model is stable for data fitting and prediction after the number of settings exceeds 45. In conclusion, the number of topics is set to 65, and all words in the corpus are transformed into the form of $< wordID : topicID >$ after LDA training.

6.4.1 Validity verification of the improved word2vec model based on TWE

The output of word2vec and the output of TWE are compared as input of CNN-VAE to verify the performance of the improved word2vec model based on TWE. The experimental results are shown in Column 5 in Tables 2, 3, and 4. Columns 4, 5 in Table 2 reveal the classification results of the two models as pretraining under KNN classification algorithm. TWE+CNN-VAE is higher than CNN-VAE in recall, precision, and F1-score, only 0.45% lower in accuracy. Columns 4, 5 in Table 3 show that the results of the four evaluation metrics of TWE+CNN-VAE are higher than those of CNN-VAE in the RF classifiers. Columns 4, 5 in Table 4 show that the accuracy and F1-score of TWE+CNN-VAE are higher than CNN-VAE in the SVM classifiers. The experimental results manifest that proposed method has better performance in typical classification algorithms and solves the polysemy problem to a certain extent. Polysemy is a common problem in natural language, especially for computer processing. The proposed method considers the topical information of a word and codes it by different vectors in different contexts which is beneficial for NLP.

Table 1 Cnews dataset

Category	Training set	Testing set	Validation set
Current affairs	5000	1000	500
Education	5000	1000	500
Entertainment	5000	1000	500
Estate	5000	1000	500
Fashion	5000	1000	500
Finance and economics	5000	1000	500
Games	5000	1000	500
Home furnishing	5000	1000	500
Science and technology	5000	1000	500
Sports	5000	1000	500
Total	50000	10000	5000

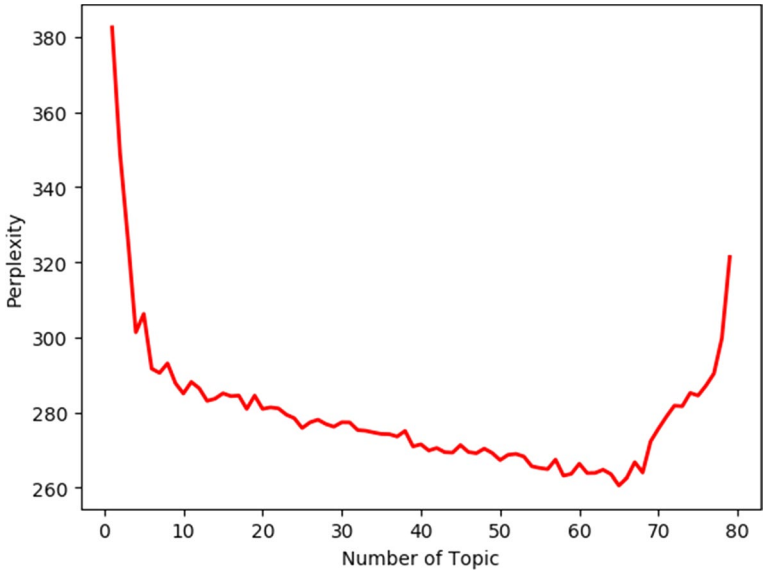


Fig. 12 The relationship between the degree of perplexity and the number of topics

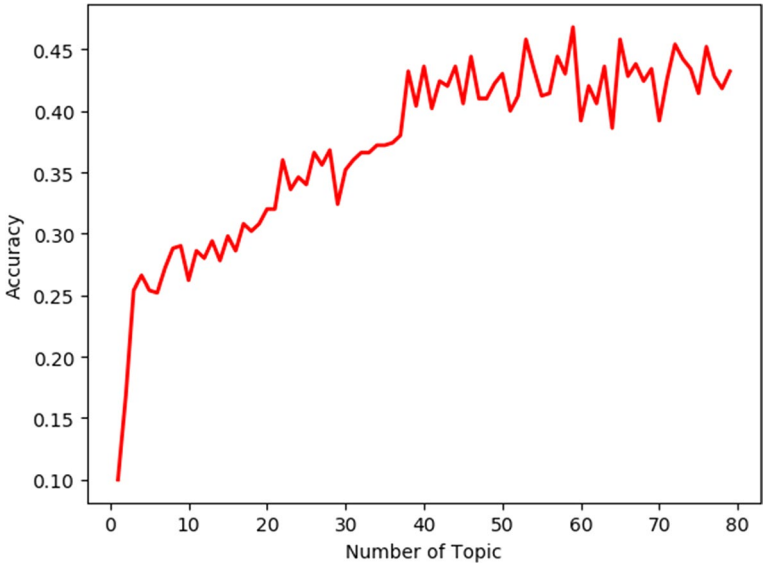


Fig. 13 The relationship between the accuracy and the number of topics

6.4.2 Comparing experiments with word2vec-avg and CNN-AE

In this paper, the proposed model is compared with two unsupervised text representation methods, which transform word vectors into document vectors. The

Table 2 Performance comparison among [18, 39], CNN-VAE, and TWE+CNN-VAE in KNN classifier

	Word2vec-avg [18]	CNN-AE [39]	CNN-VAE	TWE+CNN-VAE
Recall	77.02 ± 7.31	86.44 ± 10.69	91.81 ± 7.46	**91.83 ± 7.45**
Accuracy	87.32 ± 7.40	92.94 ± 5.06	**94.87 ± 4.61**	94.42 ± 5.27
Precision	80.82 ± 7.61	89.56 ± 11.12	91.87 ± 9.12	**93.34 ± 7.75**
F1-Score	77.28 ± 7.38	86.66 ± 10.70	90.87 ± 8.08	**91.47 ± 7.24**

Bold data represents the best result

Table 3 Performance comparison among [18, 39], CNN-VAE, and TWE+CNN-VAE in RF classifier

	Word2vec-avg [18]	CNN-AE [39]	CNN-VAE	TWE+CNN-VAE
Recall	66.23 ± 2.27	80.26 ± 1.42	92.54 ± 1.67	**92.76 ± 1.09**
Accuracy	85.48 ± 0.93	92.43 ± 0.66	94.98 ± 0.65	**96.85 ± 0.51**
Precision	74.84 ± 4.25	92.64 ± 3.18	95.29 ± 1.00	**96.65 ± 0.46**
F1-Score	68.27 ± 2.73	83.97 ± 1.85	93.37 ± 1.22	**94.92 ± 0.79**

Bold data represents the best result

Table 4 Performance comparison among [18, 39], CNN-VAE, and TWE+CNN-VAE in SVM classifier

	Word2vec-avg [18]	CNN-AE [39]	CNN-VAE	TWE+CNN-VAE
Recall	87.61 ± 3.08	90.94 ± 1.48	**93.90 ± 2.08**	93.87 ± 3.21
Accuracy	86.51 ± 3.18	90.00 ± 1.64	93.30 ± 2.63	**93.49 ± 3.53**
Precision	88.68 ± 2.83	90.10 ± 2.98	**94.29 ± 1.90**	94.23 ± 2.77
F1-Score	87.58 ± 3.27	90.64 ± 1.55	93.83 ± 1.99	**94.05 ± 3.20**

Bold data represents the best result

comparison models can be defined as follows.

- word2vec-avg: This model obtains the last representation by averaging all word vectors of each document. In this experiment, the training word vector dimension and the text representation vector dimension are set to 128 [18].
- CNN-AE: This is a CNN-based AE model. The word vectors gained by CNN can be used as the input of encoder to get the text representation [39].

In Columns 2,3 in Tables 2, 3, and 4, the performance of word2vec-avg is inferior to CNN-AE. It shows that CNN is good at fetching the local semantic features among words. Besides, the CNN-VAE model based on TWE outperforms CNN-AE and word2vec-avg in KNN, RF, and SVM classifiers when the text representation dimension is 128. In other words, if VAE is used instead of AE, the vector distribution of text feature space can more conform to the normal distribution, so that the semantic information can be in line with the actual distribution. In summary, the CNN-VAE text feature representation model based on TWE is reasonable and

effective. To show the performance difference more intuitively, this paper draws the histograms of the performance of the four models as shown in Figs. 14, 15, and 16.

7 Conclusions and future work

Text feature extraction is a learning algorithm of data preprocessing, which can not only effectively reduce the dimension of feature space but also directly affect the accuracy of subsequent classification algorithms. Besides, polysemy in natural language is also worthy of attention. Polysemy means that the same word can express multiple meanings, which is the general ambiguity in natural language. To effectively extract the semantic features among words and distinguish polysemy in natural language, this paper proposes an intelligent CNN-VAE text representation algorithm as an advanced learning method for social big data within next-generation IIoT, which helps users to identify the information collected by sensors and perform further processing. This method employs the convolution layer to capture the local features of the context and the variational technique to reconstruct feature space to make it conform to the normal distribution. In addition, the output of the improved word2vec model based on TWE is employed as the input of the proposed model to distinguish polysemy. The paper utilizes Cnews dataset to verify the performance of the proposed method with four evaluating metrics (i.e., recall, accuracy, precision, and F1-score). Experimental results indicate that the proposed method outperforms word2vec-avg and CNN-AE in KNN, RF, and SVM classifiers and distinguishes polysemy effectively.

With the improvement of hardware computing performance and the development of deep learning, text representation technology has developed by leaps and bounds, and more effective NLP models have been proposed. Accordingly, our future work includes: 1) improving the performance of these models in text feature representation and 2) applying these models in different levels of language structures.

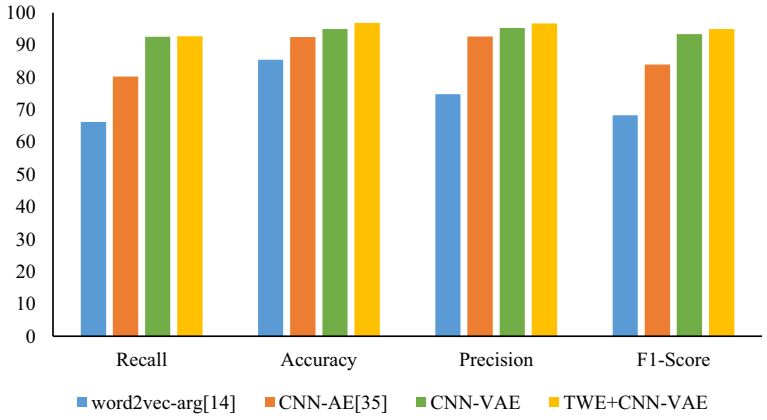


Fig. 14 Histogram of performance comparison among [18, 39], CNN-VAE, and TWE + CNN-VAE in KNN classifier

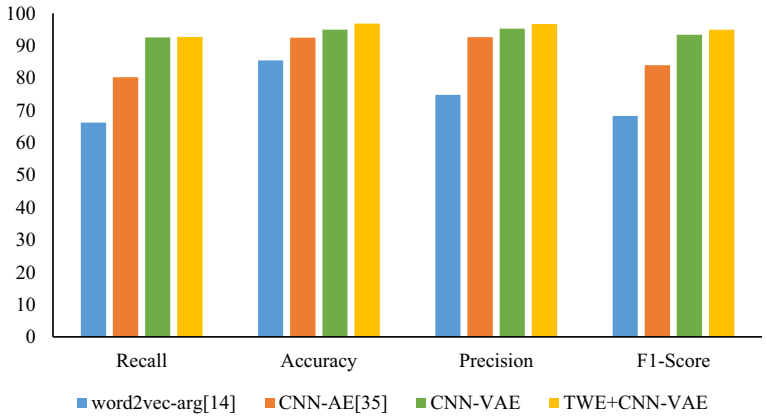


Fig. 15 Histogram of performance comparison among [18, 39], CNN-VAE, and TWE + CNN-VAE in RF classifier

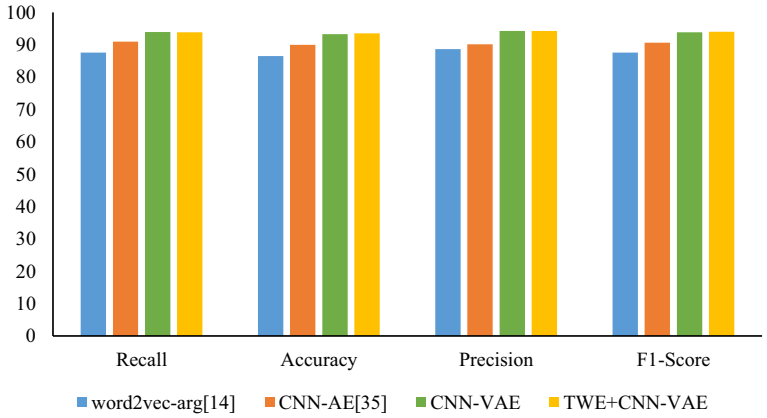


Fig. 16 Histogram of performance comparison among [18, 39], CNN-VAE, and TWE + CNN-VAE in SVM classifier

Author Contributions Saijuan Xu contributed to conceptualization, methodology, investigation, and writing—original draft. Canyang Guo was involved in data curation, validation, methodology, and writing—original draft. Yuhan Zhu contributed to methodology, investigation, visualization, writing—review and editing, and provided software. Genggeng Liu was involved in resources, supervision, and writing—review and editing. Neal Xiong contributed to formal analysis, supervision, and writing—review and editing.

Funding This work was partially supported by the Natural Science Foundation of Fujian Province under Grant No. 2019J01243.

Availability of data and materials The paper uses the open dataset Cnews. Available: http://thuctc.thunlp.org/.

Declarations

Ethics approval We certify that all authors have seen and approved the final version of the manuscript being submitted. We warrant that the article is the original work, and it has not received prior publication and is not under consideration for publication elsewhere.

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Guo W, Shi Y, Wang S, Xiong NN (2020) An unsupervised embedding learning feature representation scheme for network big data analysis. IEEE Trans Netw Sci Eng 7(1):115–126
2. Jiang D, Huo L, Song H (2020) Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans Netw Sci Eng 7(1):80–90
3. Chen Y, Zhou L, Pei S, Yu Z, Chen Y, Liu X, Du J, Xiong N (2019) KNN-BLOCK DBSCAN: fast clustering for large-scale data. IEEE Trans Syst Man Cybern Syst 51(6):3939–3953
4. Cheng H, Xie Z, Shi Y, Xiong N (2019) Multi-step data prediction in wireless sensor networks based on one-dimensional CNN and bidirectional LSTM. IEEE Access 7:117883–117896
5. Yao Y, Xiong N, Park JH, Ma L, Liu J (2013) Privacy-preserving max/min query in two-tiered wireless sensor networks. Comput Math Appl 65(9):1318–1325
6. Liu N, Zhang B, Yan J, Chen Z, Liu WY, Bai FS, Chien LF (2005) Text representation: from vector to tensor. In: IEEE International Conference on Data Mining
7. Liu GZ (2010) Semantic vector space model: implementation and evaluation. J Assoc Inform Ence Technol 48(5):395–417
8. Liu Y, Li K, Yan D, Gu S (2022) A network-based CNN model to identify the hidden information in text data. Phys A Stat Mech Appl 590:126744
9. Mewada A, Dewang RK (2022) SA-ASBA: a hybrid model for aspect-based sentiment analysis using synthetic attention in pre-trained language bert model with extreme gradient boosting. J Supercomput, 1–36
10. Zhang Y, Jin R, Zhou Z (2010) Understanding bag-of-words model: a statistical framework. Int J Mach Learn Cybern 1(1–4):43–52
11. Yan D, Li K, Gu S, Yang L (2020) Network-based bag-of-words model for text classification. IEEE Access 8:82641–82652. https://doi.org/10.1109/ACCESS.2020.2991074
12. Garcia D, Hu X, Rohrer M (2023) The colour of finance words. J Financ Econ 147(3):525–549
13. Zhu J, Fang Y, Yang P, Wang Q (2016) Research on text representation model integrated semantic relationship. In: IEEE International Conference on Systems
14. Liang H, Sun X, Gao Y (2017) Text feature extraction based on deep learning: a review. Eurasip J Wireless Commun Netw 2017(1):211
15. Young T, Hazarika D, Poria S, Cambria E (2018) Recent trends in deep learning based natural language processing [review article]. IEEE Comput Intell Magazine 13(3):55–75
16. Gao Y, Xiang X, Xiong N, Huang B, Lee HJ, Alrifai R, Jiang X, Fang Z (2018) Human action monitoring for healthcare based on deep learning. IEEE Access 6:52277–52285
17. Petrovic D, Janicijevic S (2019) Domain specific word embedding matrix for training neural networks. In: 2019 International Conference on Artificial Intelligence: Applications and Innovations, pp. 71–714
18. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inform Process Syst 26:3111–3119
19. Alshari EM, Azman A, Doraisamy S, Mustapha N, Alksher M (2020) Senti2vec: an effective feature extraction technique for sentiment analysis based on word2vec. Malays J Comput Sci 33(3):240–251
20. Ji S, Yun H, Yanardag P, Matsushima S, Vishwanathan SVN (2016) Wordrank: learning word embeddings via robust ranking. Comput Ence, 658–668

21. Hill F, Cho K, Korhonen A (2016) Learning distributed representations of sentences from unlabelled data. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies
22. Xie F, Wen H, Wu J, Chen S, Hou W, Jiang Y (2019) Convolution based feature extraction for edge computing access authentication. IEEE Trans Netw Sci Eng 7:2336–2346
23. Nie L, Ning Z, Wang X, Hu X, Li Y, Cheng J (2020) Data-driven intrusion detection for intelligent internet of vehicles: a deep convolutional neural network-based method. IEEE Trans Netw Sci Eng 7:2219–2230
24. Hu B, Lu Z, Li H, Chen Q (2015) Convolutional neural network architectures for matching natural language sentences. In: Proceedings of Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems
25. Jang B, Kim I, Kim JW (2019) Word2vec convolutional neural networks for classification of news articles and tweets. PloS One 14(8):0220976
26. Kalchbrenner N, Grefenstette E, Blunsom PA (2014) A convolutional neural network for modelling sentences. Eprint Arxiv, 1
27. Hao Z, Yeh W, Hu C, Xiong NN, Su Y, Huang C (2020) A novel convolution-based algorithm for the acyclic network symbolic reliability function problem. IEEE Access 8:99337–99345
28. Kido S, Hirano Y, Hashimoto N (2018) Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN). In: 2018 International Workshop on Advanced Image Technology, pp. 1–4
29. Shu G, Liu W, Zheng X, Li J (2018) If-CNN: Image-aware inference framework for CNN with the collaboration of mobile devices and cloud. IEEE Access 6:68621–68633
30. Yin W, Schutze H (2015) Convolutional neural network for paraphrase identification. In: Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 901–911
31. Luan Y, Lin S (2019) Research on text classification based on CNN and LSTM. In: 2019 IEEE International Conference on Artificial Intelligence and Computer Applications. pp. 352–355
32. Shi M, Wang K, Li C (2019) A c-lstm with word embedding model for news text classification. In: 2019 IEEE/ACIS 18th International Conference on Computer and Information Science, pp. 253–257
33. Bai X (2018) Text classification based on LSTM and attention. In: 2018 Thirteenth International Conference on Digital Information Management, pp. 29–32
34. Li C, Zhan G, Li Z (2018) News text classification based on improved bi-LSTM-CNN. In: 2018 9th International Conference on Information Technology in Medicine and Education, pp. 890–893
35. Li P, Chen Z, Yang LT, Gao J, Zhang Q, Deen MJ (2018) An improved stacked auto-encoder for network traffic flow classification. IEEE Netw. 32(6):22–27
36. Ameur H, Jamoussi S, Hamadou AB (2018) A new method for sentiment analysis using contextual auto-encoders. J Comput Sci Technol 33(6):1307–1319
37. Lu G, Zhao X, Yin J, Yang W, Li B (2018) Multi-task learning using variational auto-encoder for sentiment classification. Pattern Recogn Lett 132:115–122
38. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114
39. Yu J, Zhou X (2020) One-dimensional residual convolutional autoencoder based feature learning for gearbox fault diagnosis. IEEE Trans Indus Inform 16(10):6347–6358
40. Hoffman MD, Blei DM, Bach FR (2010) Online learning for latent dirichlet allocation. In: International Conference on Neural Information Processing Systems
41. Guven ZA, Diri B, Cakaloglu T (2018) Classification of new titles by two stage latent dirichlet allocation. In: 2018 Innovations in Intelligent Systems and Applications Conference, pp. 1–5
42. Kanungsukkasem N, Leelanupab T (2019) Financial latent dirichlet allocation (finlda): feature extraction in text and data mining for financial time series prediction. IEEE Access 7:71645–71664
43. Sohail AS, Sameen M, Ahmed Q (2019) Latent dirichlet allocation algorithm using linguistic analysis. In: 2019 International Conference on Green and Human Information Technology, pp. 116–118
44. DiMaggio P, Nag M, Blei D (2013) Exploiting affinities between topic modeling and the sociological perspective on culture: application to newspaper coverage of us government arts funding. Poetics 41(6):570–606
45. Hsu C, Chiu C (2017) A hybrid latent dirichlet allocation approach for topic classification. In: 2017 IEEE International Conference on Innovations in Intelligent Systems and Applications, pp. 312–315
46. Novichkova S, Egorov S, Daraselia N (2003) Medscan, a natural language processing engine for medline abstracts. Bioinformatics 19(13):1699–1706

47. Wu Q, Kuang Y, Hong Q, She Y (2019) Frontier knowledge discovery and visualization in cancer field based on kos and lda. Scientometrics 118(3):979–1010
48. Liu Y, Liu Z, Chua TS, Sun M (2015) Topical word embeddings. In: National Conference on Artificial Intelligence, pp. 2418–2424
49. Bordes A, Glorot X, Weston J, Bengio Y (2014) A semantic matching energy function for learning with multi-relational data. Mach Learn 94(2):233–259
50. Hu Z, Yang Z, Liang X, Salakhutdinov R, Xing E (2017) Controllable text generation. ArXiv
51. Hershey JR, Olsen PA (2007) Approximating the kullback leibler divergence between gaussian mixture models. In: IEEE International Conference on Acoustics
52. Liu G, Guo C, Xie L, Liu W, Xiong N, Chen G (2020) An intelligent cnn-vae text representation technology based on text semantics for comprehensive big data. arXiv preprint arXiv:2008.12522
53. Kim C, Nelson CR (2000) State-space models with regime-switching: classical and Gibbs sampling approaches with applications. J Am Stat Assoc 95(452):1373
54. Xiao Y, Li B, Gong Z (2018) Real-time identification of urban rainstorm waterlogging disasters based on weibo big data. Nat Hazards 94(2):833–842
55. RSS subscription channel of Sina news. [Online]. Available: http://rss.sina.com.cn/news/

Authors and Affiliations

Saijuan Xu[1] · Canyang Guo[2] · Yuhan Zhu[2] · Genggeng Liu[2] · Neal Xiong[3]

Saijuan Xu
xusaijuan@fjbu.edu.cn

Canyang Guo
canyangguo@163.com

Yuhan Zhu
zz_yuhan@163.com

Neal Xiong
xiongnaixue@gmail.com

[1] College of Information Engineering, Fujian Business University, Lianpan Road No.2, Fuzhou 350506, Fujian, China

[2] College of Computer and Data Science, Fuzhou University, Xueyuan Road No.2, Fuzhou 350116, Fujian, China

[3] Department of Computer, Mathematical and Physical Sciences, Sul Ross State University, 1404 East Highway 90, Alpine, TX 79830, USA