ISSN: 1813-4890

A Short Text Clustering Model for User Online Comments

Dianjie Bi [a], Houjun Liang [b], Xiaoling Chen [c] and Yonghong Yu [d]

School of Management Science and Engineering, Anhui University of Finance and Economics, Bengbu 233030, China;

[a]bidianjie@126.com, [b]lhj0013@126.com, [c]pianran2004@163.com, [d]ac120107@163.com

Abstract

Aiming at the characteristics of short comment text length and irregular writing, a K-means++ short text clustering model (CNN_KMCPP model) based on convolutional neural network is proposed.Firstly, the model denoises the obtained comment text set data, and then uses the Doc2vec model to convert the text into a sentence vector expression form; Then, in order to improve the clustering effect, convolutional neural network is used to reduce the dimension of the text and obtain more accurate text feature vectors. Finally, K-means++ clustering algorithm is used to cluster text features. The experimental results of the user review text set composed of four types of comments show that the clustering accuracy of the CNN_KMCPP model proposed in this paper has reached 0.767 and the F1 value reaches 0.809, which preliminarily shows the technical feasibility of the model.

Keywords

Comment text; Clustering; Convolutional neural network; Doc2vec model; K-means++.

1. Introduction

A large number of netizens post online comments on social media platforms such as Baidu Tieba, Bilibili, and Zhihu, thus generating massive short text data.How to quickly and effectively cluster these massive short texts has great practical value in many application scenarios.

Text clustering is an unsupervised machine learning method in the field of natural language processing. It can divide text into different cluster classes without requiring prior annotation of text categories. Traditional text clustering algorithms fall into the following categories: clustering algorithm based on division, clustering algorithm based on hierarchy, clustering algorithm based on density, clustering algorithm based on grid and clustering algorithm based on model [1].

The research on text clustering began in the 1980s, but the research on short text began with the emergence of social platforms and online comments. Different from the ordinary long text, short text has its remarkable characteristics. Firstly, when online users post comments, their writing is not standardized and misspellings are common; The second characteristic is the short length of the text, which leads to sparse text features. Because of these characteristics of short text, the traditional text clustering algorithm is not effective to cluster short text.

To improve clustering performance, most scholars start from solving the problem of sparse short text features and provide solutions from different technical perspectives. Literature [2] proposes an online semantic enhancement graphical model for evolving short text stream clustering, which automatically extracts evolving topics in the subspace of term change online. This model can effectively process large-scale data streams, and compared with literature [3-4], this model does not need to rely on external knowledge base, and can avoid the disadvantages brought by the subjective selection of external knowledge base.Literature [5] proposed a deep embedding method for feature extraction and clustering allocation using a

ISSN: 1813-4890

sentence distributed embedding autoencoder. This method maps the data space to a low dimensional feature space and iteratively optimizes the clustering objectives. In the literature [6], for the sparse vector representation of short texts, autoencoder model and sentence embedding technology are also used to conduct in-depth research on the learning of predictor features.Literature [7] proposed a solution named VEPH. In the first stage, the original text vector is projected onto a low-dimensional space, and documents located in the same one-dimensional space are grouped into the same cluster; The second stage cleans up class clusters by removing all dissimilar elements, and then iteratively merges similar class clusters in a hierarchical aggregation manner. The experimental results demonstrate the superiority of VEPH compared to other relevant literatures.

In order to improve clustering performance, a small number of scholars have proposed clustering algorithms based on feature string matching starting from the non-standard writing of comment short texts. Literature [8] proposed a clustering algorithm based on feature string matching for irregularly written mutated short texts, which is simple and efficient for clustering mutated short texts. Literature [9] suggests that despite spelling errors and high data noise in short texts, short texts on the same topic often have the same or similar phrases. Based on this, the author proposes a feature extraction algorithm based on repeat strings. The core idea is to extract the key repeat strings with complete semantics from the text set, and at the same time to count the frequency of the key repeat strings.Based on this, the authors of the literature proposed a feature extraction algorithm based on repeated strings, with the core idea of extracting semantically complete key repeated strings from the text set and calculating the frequency information of key repeated strings. Compared to traditional algorithms, this algorithm has improved its effectiveness and efficiency in short text clustering.

Due to the inability of ordinary short text clustering algorithms to capture the contextual semantics of words, more and more researchers are attempting to use deep learning models to complete short text clustering, and achieve good experimental results.Literature [10] designed a new method of short text feature extraction based on deep learning technology. It uses two-way long and short time memory network to mine the semantic information of the preceding and subsequent text, and extends the mined semantic of the preceding and subsequent text into the word vector of the central word, which is used as the vector representation of the short text for subsequent feature extraction.In the literature [11], the author embedded words into the convolutional neural network to learn the deep feature representation, and used output units to fit the pre-trained binary code in the training process. Finally, K-means was used to cluster the learned representation.Literature [12] proposes an automatic clustering algorithm of the same semantic feature words based on combinatorial neural network for short public opinion text, which effectively improves the accuracy of the representation model construction of short public opinion text.The core idea is to automatically cluster segmented public opinion short texts into feature word clusters, and then select word clusters that can represent public opinion short texts based on the frequency of feature word clusters to represent public opinion short texts.

Considering the deep learning characteristics of convolutional neural networks, based on previous work, this paper proposes a K-means++ short text clustering model based on convolutional neural networks. The convolutional neural network is used to extract deep semantic features of text, and then the K-means++ algorithm is used for text clustering. The experimental results show that the model is feasible.

2. K-means++ clustering model based on convolutional neural network

The workflow of the K-means++ short text clustering model based on convolutional neural network proposed in this paper (CNN_KMCPP model,see Fig. 1) is as follows: First, the comment

ISSN: 1813-4890

text is crawled from the network, then the comment text is cleaned to remove the noise data, and use Doc2vec to vectorization the cleaned comment text; Then, the text vector is input into the convolutional neural network to obtain the deep features of the comment text; Finally, K-means++ is used to cluster the comment text.

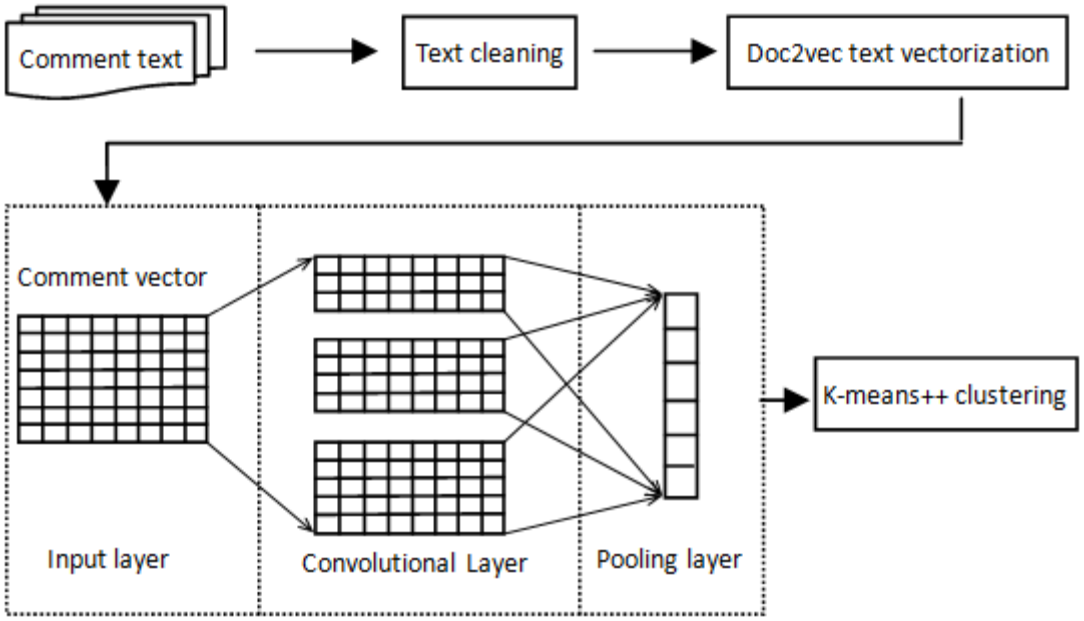

Fig.1 CNN_KMCPP model

2.1. Text cleaning

The purpose of data cleaning is to improve the quality of clustering, as the crawling network comment text data contains some noisy data, such as "sofa", "passing by" and other cliche comments, emoticon comments, spam comments, and ad replies unrelated to the theme content, etc. [13], all of which belong to noise data; The existence of these noisy data will affect the quality of clustering, so it is necessary to delete these comments.

This article adopts the "water stick filtering" algorithm proposed in literature [13] to denoise comment text. The core idea of the algorithm is to consider two factors simultaneously during denoising: the number of co-occurrence words and the length of comment text. The algorithm steps are as follows:

Step 1. If more than 2 meaningless words appear in the comment text (sofa, leave a name, follow, pass by, etc.), it will be considered as noise and deleted;

Step 2. Delete comments that are pure expressions and symbols;

Step 3. Calculate the number of co-occurrence words in the comment text, denoted as n. If n>0, the comment will be regarded as noise and deleted;

Step 4. Otherwise, if n<=0, calculate the comment text length len, denoted as len; if len>=150, the comment is regarded as noise and deleted.

2.2. Text Vector Representation Based on Doc2vec

Text vector representation is the basis of text clustering.The text after cleaning pretreatment needs to be further vectorization. Word2vec and Doc2vec are commonly used text vectorization representation methods. The Doc2vec method was proposed by Tomas Mikolov based on Word2vec [14]. Compared with Word2vec, Doc2vec adds a sentence vector that shares the weight of the same sentence in different training sessions. This means that the context information of the sentence is fully utilized when predicting the probability of a certain word,

ISSN: 1813-4890

overcoming the shortcomings of Word2vec model which cannot obtain word order information,which is important for comment texts.

In this paper, the PV-DM model of Doc2vec is used to complete text vectorization.The structure of PV-DM model, see Fig. 2.

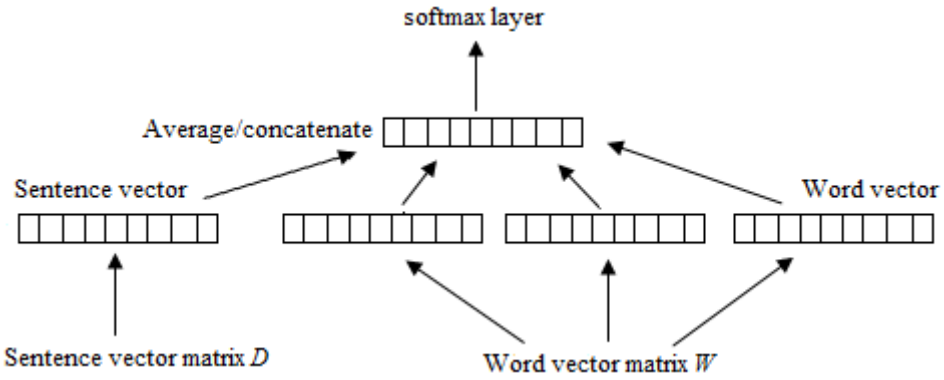

Fig.2 PV-DM Model

In Figure 2, each column of matrix *D* is the sentence vector *Di*, and each column of matrix *W* is the word vector *Wi*. *Di* and *Wi* have the same dimensions; After averaging or concatenating *Di* and *Wi*, they are used as inputs to the softmax layer to predict the probability of the target word. The model is trained using random gradient descent and backpropagation, and the training results are used as input vectors for the convolutional neural network.

2.3. Feature Extraction Based on Convolutional Neural Network

The convolutional neural network consists of an input layer, convolutional layers, pooling layers, and a fully connected layer. The output results of PV-DM model of Doc2vec are also used as the sentence vector matrix of the input layer of convolutional neural network.

2.3.1. Convolutional layer

In the convolutional neural network, the function of convolutional layers is to extract the feature information in the input text by sliding the convolutional window.The different sizes of convolutional windows mean that the local features of the obtained text are also different. The size of the convolution window is determined by the width and height of the convolution kernel. The width of the convolution kernel is consistent with the dimension of the text vector, and the height of the convolution kernel can be set independently. In this paper, multiple convolutions of different sizes are used to verify the text for convolution operations.

The sentence vector is denoted as $X \in D^{n\times d}$, where D is the sentence vector matrix, n is the number of sentence vectors, and d is the vector dimension. The result of the convolution operation is denoted as c, then the convolution operation is:

$$c_j = \mathrm{ReLU}(X_{j:j+m-1} + b) \tag{1}$$

In this formula, c_j is the value at the JTH position of vector c, and the range of values for j is: $1 \le j \le n-m+1$, m is the height of the convolution kernel, b is the offset term, and ReLU is the nonlinear activation function.

A feature map obtained by the vector element $[X_1, X_2, \ldots, X_m]$ with length m after convolution operation, that is to say, it is mapped to a new feature vector c.

$$c = [c_1, c_2, \ldots, c_m] \tag{2}$$

2.3.2. Pooling layer

The function of the pooling layer is to perform secondary extraction on the feature vectors extracted from the convolutional layer, selecting stronger expressive feature vectors as vector

ISSN: 1813-4890

representations of the text. There are various pooling methods, and This paper uses segmental pooling. Its main idea is to cut the sentence vector into several segments based on the sentence transition word, and extract a maximum feature value from each segment [15] , so that multiple local eigenvalues can be retained.

The feature vector c_j obtained through convolutional kernel operation is divided into k parts using a segmented pooling strategy: $c_{j1}, c_{j2}, \ldots, c_{jk}$. The feature values s filtered by the pooling layer can be expressed as:

$$s_{ji} = max(c_{ji}) \tag{3}$$

In this formula, $1 \le i \le k$ and $1 \le j \le m$

2.3.3. Connection layer

The function of the connection layer is to concatenate all local feature vectors obtained after pooling operations to obtain a complete feature map $c_j^{'}$ of the comment text.

$$C_j' = s_{j1} \oplus s_{j2} \oplus \cdots \oplus s_{jk} \tag{4}$$

2.4. K-means++ clustering

The K-means algorithm is widely used in large text clustering because of its simplicity and efficiency. However, due to subjective factors such as determining the number of clusters and selecting the initial cluster center point, the clustering results are prone to falling into local optima. Based on this, K-means++ algorithm optimized the selection of initial points for clustering centers,so that the distance between the initial clustering centers should be as far as possible, and the problem of local optimal solution is overcome.In this paper, K-means++ clustering algorithm is used to cluster the text feature vectors output by convolutional neural networks.

The K-means++ text clustering algorithm process is as follows:

Step 1. Select vector $c_i^{'}$ from the feature vector set $C^{'} = \{c_1^{'}, c_2^{'}, \ldots, c_n^{'}\}$ of the short text as the first clustering center;

Step 2. Calculate the distance between each other feature vector in vector set $C^{'}$ and $c_i^{'}$, denoted as $d_i = \sqrt{(c_j^{'} - c_i^{'})^2}$, and select the text feature vector with the highest value of d_i as the next clustering center based on probability $d_i / \sum d_i$;

Step 3. Repeat step 2 until *K* cluster centers are selected;

Step 4. According to the principle of being closest to the center point, divide the remaining feature vectors in $C^{'}$ into corresponding clusters;

Step 5. Use the geometric centers of each cluster as their new clustering centers;

Step 6. Repeat steps 4 and 5 until the algorithm converges and the clustering center stabilizes.

3. Experimental Analysis

3.1. Experimental Short Text Set

The authors used the web crawler program to crawl the football comment data from the Manchester United Bar of Baidu Post Bar. After data cleaning, there were 1934 pieces in total; Crawling computer review data from the laptop section of JD.com, a total of 6250 comments were obtained after data cleaning; Crawling current political comment data from the international news section of Sina. com, a total of 6010 comments were obtained after data cleaning; Crawling music comment data from the music section of Station B, after data cleaning, a total of 7480 comments were obtained. There were a total of 21674 comment texts in four

ISSN: 1813-4890

categories: football comments, computer comments, political comments, and music comments, which were randomly scrambled to form an experimental short text dataset. The sample data , see Table 1.

Table 1 Sample comment text

Numble	Comment content	Category (without label)
1	The last one who was knocked down by two people in a row was a penalty kick. (In Chinese)	Football Review
2	There is a solemn sense of time and space travel and dislocation. (In Chinese)	Music Review
3	This is a machine player with very strong abilities. (In Chinese)	Football Review
4	The tone has a feeling of being out of the world, a bit desolate but beautiful. (In Chinese)	Music Review
5	The startup speed is very fast, and it starts in 5 seconds. Screen effect: The screen is very high-definition, watching videos, working, and the eyes are not tired at all. (In Chinese)	Computer comments
6	Easy to go out, difficult to come in! You must obtain the consent of all members of UNESCO, otherwise you will definitely not be able to enter! (In Chinese)	Political commentary
7	The resumption of diplomatic relations between Iran and Saudi Arabia is what all peace loving people in the world hope for. (In Chinese)	Political commentary
8	Sound very good! Show the importance and urgency of the original song to the point, with full emotions but no affectation, and very comfortable fingering! (In Chinese)	Music Review

3.2. Cluster evaluation indicators

Because the number of categories in the comment text set of this article and the number of texts contained in each category are known, three external evaluation indicators such as accuracy, recall, and F1 value are used to determine the quality of the clustering results. The accuracy represents the accuracy of clustering results, as shown in formula (5); The recall rate indicates whether all samples of the cluster can be found, as shown in formula (6); The F1 value combines the results of accuracy and recall, and can comprehensively evaluate the accuracy and recall. When F1 is higher, it indicates that the test method is more effective, as shown in formula (7).

$$Precision = \frac{N_{i,j}}{N_j} \tag{5}$$

$$Recall = \frac{N_{i,j}}{N_i} \tag{6}$$

$$F_1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

In the formula, N_i represents the number of texts contained in category *i*, N_j represents the total number of texts contained in all categories in cluster *j* , and $N_{i,j}$ represents the number of categories *i* contained in cluster *j*.

ISSN: 1813-4890

3.3. Neural network parameters setting

In this paper, the convolutional neural network is built based on PyTorch, and the model parameters are set as shown in Table 2. The size of the convolution kernel is set to (3,4,5), and the sentence vector dimension is set to 150. The learning rate is an important hyperparameter that controls the learning speed of the model; If it is too small, it will slow down the network convergence and may not be able to find the optimal solution; If it is too large, the network may not converge, causing the loss function to oscillate significantly, and the optimal solution may be skipped;After multiple experiments, setting the learning rate to 0.5 is more appropriate. To improve the execution speed of the model, the sample subset mini-batch is set to 100. In order to accelerate the convergence speed and solve the problem of gradient disappearance, ReLU is selected as the activation function of the model. Too many iterations of the model may lead to overfitting, and too few iterations will not achieve the required accuracy. The authors finally selects the number of training iterations epoch=30.

Table 2 CNN model parameter settings

parameter	Attribute value
Convolutional kernel	3,4,5
Vector dimension	150
Learning rate	0.5
mini-batch	100
epoch	30
Activation function	Relu

3.4. Experimental analysis and comparison

In order to verify the effect of CNN_KMCPP model on short-text clustering, the authors selects two representative models: the short-text clustering model based on LDA and the short-text clustering model based on Word2vec word vector, and makes experimental comparison with the short-text clustering model proposed in this paper on the short-text set obtained in Section 3.1. The results are shown in Table 3.

Table 3 Comparison of evaluation results

model	Accuracy	recall	F1 value
LDA model	0.694	0.753	0.722
Wor2Vec Word Vector Model	0.713	0.825	0.765
CNN_ KMCPP model	0.767	0.858	0.809

The LDA topic model can usually be used to learn the topic information hidden in large-scale text sets. However, because of the sparse features of short texts, the clustering effect of using the LDA topic model is often not ideal. The experiment on the short text set in section 3.1 also verifies this. The CNN_KMCPP model proposed in this paper uses multiple convolutional kernels and segmented extraction of deep semantic features from text, which significantly improves accuracy, recall, and F1 value compared to traditional LDA clustering models.The Word2vec word vector model is a simple three-layer neural network model that includes an input layer, a hidden layer, and an output layer. The Word2vec word vector model is essentially a bag-of-words model model. Although the semantics of words are considered, the effect of word order on the overall meaning of sentences is not considered, which has a certain impact on the clustering effect of short texts. In this paper, the Doc2vec sentence vector model is used for text representation, which fully considers the influence of the order between words in text context on text semantics. The experimental results show that compared with the Word2vec word vector model, the accuracy rate, recall rate and F1 value are also improved slightly.

In summary, the experimental results on text sets show that the CNN_KMCPP short text clustering model proposed by the authors achieves good clustering effect.

ISSN: 1813-4890

4. Conclusion

In this paper, a short text clustering model of CNN_KMCPP is proposed in view of the sparse features, short text and irregular grammar of online comments.This model mainly uses Doc2vec to represent short text vector, uses convolutional neural network to extract features, and finally uses K-means++ to cluster text. Experiments on the crawling short text set show that the accuracy rate of the CNN_KMCPP model is 0.767, the recall rate is 0.858, and the F1 value is 0.809. The validity of the model has been preliminarily verified.

The short text set used in this article includes four categories: football comments, political comments, computer comments, and music comments. The number of clusters is predetermined, and the number of text sets is relatively small. The next step is to observe the clustering effect of CNN_KMCPP model on the large text set with uncertain number of clusters.

Acknowledgements

This work was supported by the Anhui University of Finance and Economics Science Research Project (Grant No. ACKYC20048) and the Anhui Provincial Natural Science Foundation Key Project (Grant No. KJ2021A0478).

References

[1] Long Liu: Research on Text Clustering Algorithm Based on ALBERT (MS., Xiangtan University, China 2021), p.7-8. (In Chinese)

[2] Kumar Jay,Din Salah Ud,Yang Qinli,Kumar Rajesh,Shao Junming:An Online Semantic-Enhanced Graphical Model for Evolving Short Text Stream Clustering.[J]. IEEE transactions on cybernetics,2021,PP.

[3] Tatsuya Nakamura,Masumi Shirakawa,Takahiro Hara and Shojiro Nishio:Wikipedia-Based Relatedness Measurements for Multilingual Short Text Clustering.[J].ACM Trans. Asian & Low-Resource Lang. Inf. Process.,2019,18(2).

[4] C.X.Jin,H.Y.Zhou:Chinese short text clustering based on dynamic vector.Computer Engineering and Applications,Vol. 47 (2011) No.33, p.156-158. (In Chinese)

[5] Zuhua Dai,Dai Zuhua,Li Kelong,Li Hongyi and Li Xiaoting:An Unsupervised Learning Short Text Clustering Method[J]. Journal of Physics: Conference Series, Vol. 1650 (2020) No.3.

[6] Dakshnamoorthy Vinoth:A Short Text Clustering Approaches in Social Media[J]. Electrochemical Society Transactions, Vol. 107 (2022) No.1.

[7] Akritidis Leonidas,Alamaniotis Miltiadis,Fevgas Athanasios and Tsompanopoulou Panagiota:Bozanis Panayiotis. Improving Hierarchical Short Text Clustering through Dominant Feature Learning[J]. International Journal on Artificial Intelligence Tools,Vol. 31 (2022) No.5.

[8] Y.G. Huang, Ting Liu,W.X. Che and X.G. Hu:A Fast Clustering Algorithm for Abnormal and Short Texts [J]. Journal of Chinese Information Processing,Vol. 21 (2007) No.2, p.63-68. (In Chinese)

[9] J.X, Hu, H.B. Xu, Yue Liu, Bin Wang and X.Q. Cheng: Short-text Clustering Based on RePeats. The 8th National Joint Conference on computational linguistics (Najing, China, August 27-29, 2005). 2005,p.367-373 (In Chinese)

[10] H.W. Wan:Research on Short Text Clustering Ensemble via Deep Learning(MS., Dalian Maritime University, China 2020). (In Chinese)

[11] Jiaming Xu,Bo Xu,Peng Wang,Suncong Zheng,Guanhua Tian,Jun Zhao and Bo Xu: Self-Taught convolutional neural networks for short text clustering[J]. Neural Networks,Vol. 88(2017) p.22-31.

[12] Da Huo,Y.M. Zhao,L.W. Zhang,Z.L. Zhang,Y.S. Wang and L.M. Liu:Research on Dimension Reduction Method of Public Opinion Short Message Text Represention Model Based on Combinatorial Neural Network. Journal of Inner Mongolia University of Technology (Natural Science),Vol. 39 (2022) No.02, p.136-140. (In Chinese)

[13] W.H. Zhu: Research on BBS Short Text Clustering (Ph.D., Harbin Institute of Technology, China 2009), p.12-16. (In Chinese)

[14] Le Q V,Mikolov T:Distributed representations of sentences and documents[J].Computer Science, 2014 No.4,p.1188-1196.

[15] Information on http://blog.csdn.net/malefactor/article/details/51078135