Computer Science and Information Systems 21(4):1783–1800 https://doi.org/10.2298/CSIS240418062J

A Novel Deep Fully Convolutional Encoder-Decoder Network and Similarity Analysis for English Education Text Event Clustering Analysis

Zhenping Jing

School of Foreign Languages, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
jingzpz@163.com

Abstract. Education event clustering for social media aims to achieve short text clustering according to event characteristics in online social networks. Traditional text event clustering has the problem of poor classification results and large computation. Therefore, we propose a novel deep fully convolutional encoder-decoder network and similarity analysis for English education text event clustering analysis in online social networks. At the encoder end, the features of text events are extracted step by step through the convolution operation of the convolution layer. The background noise is suppressed layer by layer while the target feature representation is obtained. The decoder end and the encoder end are symmetrical in structure. In the decoder end, the high-level feature representation obtained by the encoder end is deconvolved and up-sampled to recover the target event layer by layer. Based on the linear model, text similarity is calculated and incremental clustering is performed. In order to verify the effectiveness of the English education text event analysis method based on the proposed approach, it is compared with other advanced methods. Experiments show that the performance of the proposed method is better than that of the benchmark model.

Keywords: Online social networks, Text event clustering, Deep fully convolutional encoder-decoder network, Similarity analysis, Linear model.

1. Introduction

In today's network era, with the development of the mobile Internet, information interaction has become unprecedentedly simple and fast. QQ, WeChat, Weibo, Douyin, Kuaishou and other social media have widely entered people's lives and changed people's living habits. Studies have shown that social media is more sensitive to new events than traditional media [1-3]. Therefore, it is of great significance to conduct data analysis on educational texts in social media. Among them, event clustering is an important step of event detection in social media education [4,5].

Event clustering aims to cluster text according to different features of text event features. Social media is mostly short text, and the text content is diverse and random, including more disturbing words. The traditional unsupervised clustering model is difficult to accurately extract the event features of social text, and the accuracy of event clustering results is low. Xu et al. [6] used a supervised deep neural network model to cluster social texts, which enhanced the clustering effect. However, faced with massive social texts, this method required a lot of text annotation work. Costache et al. [7] proposed a Logit model to estimate weights instead of using weights determined entirely by distance. The nearest neighbor was automatically selected using a selection process such as lasso or enhancement. Then, based on the concepts of evaluation and selection, the predictor space was expanded. Balogun et al. [8] proposed an adaptive ASMC (Adaptive Selection of Misprediction Classifiers) method, which dynamically selected a classifier based on the characteristics of the class to better predict the error tendency of a class from a set of machine learning classifiers. An empirical study of 30 software systems showed that ASMC performed better than five classifiers used alone and combined with majority vote integration [9].

At present, most event clustering studies are mainly based on word characteristics in online social networks. Different from long text, short text clustering has the problem of high dimensional sparsity [10], so some scholars consider introducing external features. Among them, Ma et al. [11] used emergent keywords to predict the importance of short texts and cluster these important short text detection events. By considering the release time, diffusion degree and diffusion sensitivity of events, Hompes et al. [12] adopted time characteristics to detect events and performed clustering. In addition, Adamopoulos et al. [13] explored the influence of users on social media data, using text content characteristics, user characteristics and usage characteristics to detect events and perform clustering. Liu et al. [14] made use of named entity features of text to enhance the effect of event detection.

In order to solve the problem of high-dimensional sparsity in traditional methods, Yang et al. [15] mapped high-dimensional text to low-dimensional semantic space through local save index (LPI), and made semantically related texts close to each other in low-dimensional space. Rezaee et al. [16] used a K-means clustering algorithm to cluster feature word sets to obtain feature clusters, and used feature clusters to represent sentence vectors, thus solving the problem of dimension explosion of vector space models and improving the clustering effect. Lilleberg et al. [17] proposed JS-IDF sequence for text embedding based on Word2Vec word vector and combined with temporal relationship. Vodrahalli et al. [18] proposed the SIF Embedding of the text, which was weighted average of the word vector and modified by PCA and SVD to obtain the low-dimensional vector representation of the text. Soni et al. [19] adopted a deep neural network based on DCNN to learn the deep feature representation of text. The model first obtained the binary encoding of the text by the existing unsupervised dimensionality reduction method, and then inputted the text into the convolutional neural network through Word Embedding (WE). The binary coding of the text was taken as the training object of the model, and the intermediate feature vector between the convolutional layer and the output layer was taken as the depth feature of the text, which was a self-training based unsupervised model.

For streaming data in social media education, common clustering algorithms include Singlepass incremental algorithm and local sensitive hash (LSH) algorithm [20]. To obtain a new sample, the Singlepass clustering algorithm first needs to calculate the similarity between the new text and the existing event. If the similarity exceeds the threshold, it will be added to the existing event with the greatest similarity; otherwise, it will be set as a new event. The key step of the algorithm is to calculate the text similarity, the most commonly used is cosine similarity cosine. The local sensitive hash (LSH) clustering algorithm is mainly based on the new event detection model. Its idea is to find the nearest neighbor text set of the new text in the clustered text through the LSH algorithm, and find the nearest neighbor text of the new text from the set. If the maximum similarity between the two is greater than the set threshold, it is an existing event; otherwise, it is a new event. In the LSH clustering algorithm, the key step is to find the nearest neighbor text of the new text by LSH algorithm as soon as possible. Zhao et al. [21] improved the hashing algorithm for finding the nearest neighbor text, which improved the efficiency, but the accuracy was comparable to Petrovic [22]. Enguehard et al. [23] proposed a deep embedded clustering model based on self-training. This model used deep neural networks to learn both feature representation and cluster assignment, and was a partition-based clustering model, which was not suitable for processing streaming data. Zhao et al. [24] used SIF Embedding to represent the sentence and used auto-encoder to extract the low-dimensional feature representation of the text. The clustering algorithm could learn both the text feature representation and cluster assignment through the self-trained neural network model to obtain the clustering result.

Pavlinek et al. [25] used the supervised SVM model to determine whether two texts were related, and conducted text clustering, which regarded the sample distribution as a graph model and realized clustering by maximizing the similarity of sample pairs within the cluster. Zhang et al. [26] used the supervised LSTM model to extract text features, calculate text similarity, and adopt the incremental clustering algorithm to cluster social text. Compared with previous clustering algorithms, the performance was improved, but a large amount of data was needed to train the model. At present, the method of fully deep convolutional neural network has not been used in the event clustering method for social education media.

Therefore, we propose a novel deep fully convolutional encoder-decoder network and similarity analysis for Chinese education text event clustering analysis in online social networks in this paper. At the encoder end, the features of text events are extracted step by step through the convolution operation of the convolution layer. Based on the linear model, text similarity is calculated and incremental clustering is performed. Experiments show that the performance of the proposed method is better than that of the benchmark model. The rest of the paper is organized as follows. The deep fully convolutional encoder-decoder network and proposed social media education events clustering model are described in Sections 2 and 3. In Section 4, we conduct experiments for this proposed model. Finally, Section 5 concludes the paper.

2. Deep Fully Convolutional Encoder-decoder Network (DFCEDN)

2.1. Convolutional Encoder-decoder Network

Encoder-decoder network is a flexible framework model, composed of encoders and decoders, which can be flexibly applied to unsupervised or supervised tasks as required, usually for network pre-training, high dimensional raw data dimensionalization, feature extraction and data compression and generation. Encoders and decoders can choose different network models such as FCN, CNN and RNN according to different tasks. In this paper, a deep fully convolutional encoder-decoder network is used to enhance text features by using the local perception and weight sharing of CNN [27,28]. The traditional neural network splices the time-frequency features of adjacent frames into a long vector as the input of the network. The change of relative position will increase the difficulty of the network to model the correlation structure in the text. Convolutional neural networks have local perception characteristics, and each neuron in the network only needs to be connected with some neurons in the previous layer. Meanwhile, the weight sharing feature of convolutional neural networks makes each filter feature map have the same weight, which will greatly reduce the number of parameters in the network and improve the robustness of the model.

The decoder end is composed of multiple convolutional layers, each of which consists of convolutional filtering, batch standardization, pooling and nonlinear transformation operations. The structure of the decoding end corresponds to that of the encoding end, which is composed of deconvolution, up-sampling, batch standardization and nonlinear transformation operation. This structure is applied to speech enhancement, and the time-frequency features of noisy speech are taken as the input of the network. The convolution network is used to model the local typical structure of noisy speech spectrum at the coding end, extract high-level speech features, and suppress the influence of background noise layer by layer. At the decoding end, the detailed speech components are recovered layer by layer from the high-level speech feature information extracted from the coding end through the deconvolution layer, and the speech signal is reconstructed.

2.2. Skip Connection

The skip connection is first introduced in neural network to solve the problem that the gradient disappears with the increase of network depth in training, which leads to the difficulty of network training. In this paper, skip connection is introduced between corresponding layers of encoder. On the one hand, the error can be directly transmitted to the lower layer, so that the training of the network is more effective, and an important role is that the details of the signal can be compensated. With the increase of the number of network layers, the main structural features of the signal will be extracted due to the operation of layer by layer, such as convolution and pooling, and some local details will be lost. Especially with the deepening of the number of network layers, it may cause the problem of recovering the details of the target signal from high-level information that cannot be obtained by the decoding end. The feature map of the coding side contains a lot of detail information of the signal. By introducing skip connections to transfer the feature map information of the coding side from the convolution layer to the corresponding deconvolution layer, it can help to recover the detail information of the target signal.

3. The Proposed Social Media Education Events Clustering Model

The clustering process of the proposed model is shown in Figure 1. For the input social media text t_i, firstly, the event cluster is carried out to determine whether t_i belongs to an existing event or a new event, or cannot determine. If t_i belongs to an existing event, it is added to the cluster. If it is a new event, a new cluster is created based on ; otherwise, t_i is set as an uncertain sample and Buffer is added. In this process, a deep fully convolutional encoder-decoder network (DFCEDN) is used to extract text features, and a linear model is used to calculate text similarity. Since the incremental clustering algorithm can obtain real-time clustering results, after each part of the data is clustered, samples are extracted from the clustering results to form a training set to retrain the model, so that the model can further learn new event features and enhance the clustering effect. At the end of the clustering, all samples in clusters with fewer elements in the clustering result are set as uncertain samples. Then, the model after repeated retraining is used to cluster the uncertain samples, and the final clustering result is obtained. The clustering process is described in detail below.

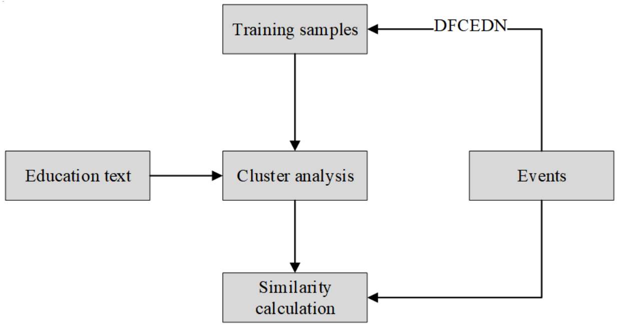

Fig. 1. Proposed education text clustering process

3.1. Text Representation

In this paper, DFCEDN model is used to extract text features, which is denoted as $M_{encoder}$. Before the text input, the first word is segmented and stopped words. The preprocessed text is represented by $X = w_1, w_2, \cdots, w_n$. Where w_i represents the number of the $i-th$ word in the sentence in the thesaurus. n represents the number of words in the sentence. The Chinese word vector pre-trained from the education entry of Baidu Encyclopedia is used to embed the sentence, and the d-dimensional word vector x_i is used to represent the words $w_i, x_i \in R^{n \times d}$. The text vector $X = x_1, x_2, \cdots, x_n$ gets a hidden sequence $h_1, h_2, \cdots, h_n$ through the DFCEDN model. Where h_t is calculated from the current input vector x_t and the output h_{t-1} of the previous time, $t \in (1, n)$, i.e.,

$$h_t = DFCEDN(x_t, h_{t-1}). \tag{1}$$

The initial parameters of the model are generated randomly, and we use a low-dimensional vector h_t to represent the text vector X.

3.2. Text Similarity Calculation

Vector similarity is a measure of the degree of similarity between vectors. If two vectors have the same starting point and the same ending point, then the two vectors are completely similar, and the similarity is equal to 1. At present, there are many researches on vector similarity, which are mainly divided into three categories. Reference [29] studied the similarity of vectors from the perspective of vector direction; Reference [30] studied the similarity of vectors from the perspective of vector size; Reference [31] studied the similarity of vectors from the perspective of vector direction and magnitude. Since a vector contains two elements: magnitude and direction, the similarity of a vector should be measured from two aspects: direction and magnitude. It is feasible to measure the similarity between two vectors from the aspects of direction and magnitude, but the position information should also be considered when measuring the similarity between multiple vectors. Therefore, we give a method to measure the similarity between multiple vectors.

The Angle cosine method is an important method to measure the similarity between two vectors, but it has a certain one-sidedness. Taking a two-dimensional vector as an example, the cosine of the Angle between vector (1,1) and vector (3,3) is equal to the cosine of the Angle between vector (1,1) and vector (0.1,0.1), but we cannot consider vector (3,3) and vector (0.1,0.1) to be exactly equivalent. Since a vector is a vector containing two elements of magnitude and direction, we should measure the similarity between two vectors in terms of direction and magnitude.

Let the vector $A = a_1, a_2, \cdots, a_n$ and vector $B = b_1, b_2, \cdots, b_n$, then the similarity between vector A and vector B can be defined as:

$$\lambda = \frac{min(||A||, ||B||)}{max(||A||, ||B||)} \cdot \frac{[A, B]}{||A|| \cdot ||B||}. \tag{2}$$

Where $||A||$ and $||B||$ represent the magnitude of vector A and vector B, respectively. $[A, B]$ represents the dot product of vector A and vector B.

When measuring the similarity between multiple vectors, we construct a maximum vector consisting of the maximum values of each component, which ensures that the other vectors are on the same side of the maximum vector. Then, the similarity between each vector and the maximum vector is measured respectively, and the similarity between these vectors is determined according to the magnitude of the similarity, which is described as follows:

Assuming that there are m vectors $X_1 = x_{11}, x_{12}, \cdots, x_{1n}$, $X_2 = x_{21}, x_{22}, \cdots, x_{2n}$, $X_m = x_{m1}, x_{m2}, \cdots, x_{mn}$. Each of these vectors has n components.

1. Constructing maximum vector $X_{max} = x_1, x_2, \cdots, x_n$, where $x_i = maxx_{1i}, x_{2i}, \cdots, x_{mi}$, $i = 1, 2, \cdots, n$. Put together the maximum values of the corresponding components of each vector.
2. The similarity of each vector X_i to the maximum vector X_{max} is calculated separately.

$$\lambda_i = \frac{||X_i||}{||X_{max}||} \cdot \frac{[X_i, X_{max}]}{||X_i|| \cdot ||X_{max}||}. \tag{3}$$

3. According to the magnitude of the similarity λ_i, we know the degree of similarity between each vector and the maximum vector. Since these vectors are on the same side of the maximum vector, the value of λ_i also reflects the degree of similarity between any two vectors.

Due to the different importance of each index to describe the essential characteristics of the research object, the weight of the index was taken into account in the cluster analysis. There is one-sidedness when the two-vector similarity calculation formula is used to measure the similarity between multiple research objects, and the logic of multi-vector similarity calculation is given. By introducing the iterative idea and using the multi-vector similarity calculation formula to accurately measure the similarity between the research objects, a new classification method is presented. The main calculation steps of cluster analysis based on multi-vector similarity are divided into the following six steps:

1. Determine the index system describing the research object. When classifying the research object, we first find out the index system that can describe the essential characteristics of the research object, and grasp the main contradiction of things. If two research objects have the same score in some major indicators, we believe that there is a high probability that they belong to the same category. Therefore, we need to carefully analyze and determine the index system describing the research objects.
2. Evaluate the research objects according to the index system. After establishing the index system of the research object, we can score and evaluate the research object according to the objective reality, and then obtain the original evaluation result matrix , which represents the data of samples containing indicators.
3. According to the consistency requirements, the original evaluation results are standardized. The order of magnitude of the subjects' scores in different indicators may vary by dozens or even hundreds of times. In addition, the direction of each index score is not necessarily the same. In reality, the higher the score of the research object in some indicators, the more it belongs to the same thing, while the lower the score of the research object in other indicators, the more it belongs to the same thing, which does not meet the consistency requirement. To this end, we need to normalize the original data by using the normalization formula of the income index or the normalization formula of the cost index, so as to eliminate the dimension of the original index, compress the evaluation result data into the interval [0,1], and ensure that the index score takes the same direction. Thus, it is helpful to calculate the similarity between the research objects, which is described as follows:
   Let $M_j = \min\limits_i y_{ij}$, $N_j = \max\limits_i y_{ij}$. Where M_j represents the minimum value of column j and N_j represents the maximum value of column j, then the normalization formula of income index and cost index are as follows.
   Normalization formula of income type index: $x_{ij} = \frac{y_{ij}-M_j}{N_j-M_j}$.
   Cost index normalization formula: $x_{ij} = \frac{N_j-y_{ij}}{N_j-M_j}$.
   According to the above formulas, we can transform the original evaluation result matrix $Y_{m\times n}$ into the standardized evaluation matrix $X_{m\times n}$.
4. The weight vector $W = w_1, w_2, \cdots, w_n$ is calculated by AHP. Combined with the standardized evaluation matrix $X_{m\times n}$, the weighted evaluation matrix $Z_{m\times n}$ is obtained. The importance of each index describing the essential characteristics of the research object is different, so the weight of the index must be considered when measuring the similarity between the research objects. AHP is a weighting method combining subjective and objective methods. We use AHP to find the weight vector $W = w_1, w_2, \cdots, w_n$, where w_i represents the weight of the $i - th$ indicator. By multiplying the weight vector by the corresponding component in the standardized evaluation matrix $X_{m\times n}$, the weighted evaluation matrix $Z_{m\times n}$ is obtained.

$$z_{ij} = x_{ij} \times w_j. \tag{4}$$

5. The vector similarity formula between multiple vectors is used to calculate the similarity between samples. The maximum component of each column in the weighted evaluation matrix $Z_{m\times n}$ is taken out and to construct one maximum vector $Z_{max} = z_1, z_2, \cdots, z_n$. Here, $z_i = max z_{1i}, z_{2i}, \cdots, z_{mi}$. On the basis of the above weighted evaluation matrix, we use the vector similarity formula between multiple vectors to calculate the similarity between each sample vector and the maximum vector.

$$\lambda_i = \frac{||Z_i||}{||Z_{max}||} \cdot \frac{[Z_i, Z_{max}]}{||Z_i|| \cdot ||Z_{max}||}. \tag{5}$$

6. Given the clustering level α, classify the research objects. After determining the clustering level α according to the historical management experience, the iterative idea is adopted to classify the research objects reasonably, which is described as follows.
   First, by comparing the similarity between the sample vector and the maximum vector and the size of the clustering level, the samples corresponding to $\lambda_i \geq \alpha$ are classified into the first class.
   Second, after removing the first class of samples, the maximum component of each column in the remaining sample vectors is extracted and a new maximum vector is constructed. Similarly, the vector similarity formula between multiple vectors is used to calculate the similarity β_i between each remaining sample vector and the new maximum vector, and the corresponding samples corresponding to $\beta_i \geq \alpha$ are classified into the second category.
   Third, repeat the above process until all samples have been classified.

3.3. Model Retraining

The feature extraction model $M_{encoder}$ and the similarity model $M_{decoder}$ are spliced together and trained simultaneously. We set an update threshold U. When U samples complete clustering, D group training data is extracted from U samples to form a training set and the model is retrained. The ratio of positive cases to negative cases in training set is 1:1. The positive example extraction method is as follows: randomly select a sample p from U samples, and then randomly select a sample q from U samples in the same cluster as p. Sample p and sample q form a group of positive examples, and set the label as 1. The extraction method of negative cases is as follows: randomly select a sample p from U samples, and then randomly select a sample q from the sample set with different clusters from p among U samples. If there are no clusters in U that are different from p, randomly select a sample q from all samples that have been clustered and are different from p, sample p and sample q form a group of negative cases, and the label is set as 0. If there are no clusters different from p in all the currently clustered samples, no training will be conducted this time.

4. Experiments and Analysis

4.1. Experiment Data

The data for this experiment came from education Weibo. A total of 10828 microblogs about 50 different education events are collected. Thirty education events are used as training data, from which samples are randomly selected to form a training set. Two texts in the same education event are positive examples, labeled 1. Two texts in different education events are negative examples, labeled 0. The ratio of positive and negative examples in the training set is 1:1, and 350000 sets of samples are extracted to train the model. The remaining 20 education event data are used as a test set, containing 2520 texts in total, for the event clustering experiment.

4.2. Experiment Model Parameter

In the clustering algorithm, the value of N is 25, and the cluster is represented by the first N texts in the event cluster. The larger N is, the stronger the representation of the event cluster is, but the amount of computation will also increase. The value range of threshold L and H in the clustering process is $0 < L < 0.5 < H < 1$. The closer the value of L is to 0, and the closer the value of H is to 1, the more stringent the selection of uncertain samples will be, but at the same time, the calculation amount will be increased. In this experiment, the value of L is 0.3 and the value of H is 0.7. The value of the update threshold U should not be too small, otherwise the training is too frequent, and the model is easily affected by some extreme values, resulting in deviation. If the U value is too large, a large number of samples can only be clustered using the original model. The U value in this experiment is 200. The larger the value of sample number D in each training set, the more obvious the learning effect of the model on event features, but at the same time, it will increase the amount of computation. At the same time, D is also affected by U. In the experiment, the number of training set samples D is 800, which is 4 times of U. The number of training rounds is 5. The experimental environment is Keras framework and Tensorflow. In the training process of feature extraction model , the pre-trained word vector is used, the vector dimension is 300, and the embedding layer is set as untrainable. The output dimension of the DFCEDN layer is 128, Dropout=0.1, Recurrent Dropout=0.1, and the rest are default parameters. The output dimension of the fully connected layer in the similarity model M is 128, the activation function is ReLu, and it also contains two Dropout layers, Dropout=0.1.

4.3. Evaluation Index

We use Purity, normalized mutual information (NMI) and adjusted Rand coefficient (ARI) as the evaluation indexes for clustering effect. Purity is the ratio of the correctly calculated number of texts to the total number of texts. Its definition is shown in equation (6).

$$Purity(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |o_k \cap c_j|. \tag{6}$$

Where N represents the total number of samples. $\Omega = o_1, o_2, \cdots, o_k$ represents the clustering cluster division obtained by the clustering model. k represents the total number of predicted event clusters. $C = c_1, c_2, \cdots, c_j$ represents the true classification. j represents the total number of predicted event clusters. The Purity range is $[0, 1]$, and the closer the Purity is to 1, the better the clustering effect is.

NMI is an entropy-based evaluation index, and its definition is shown in equation (7).

$$NMI(\Omega, C) = \frac{I(\Omega, C)}{[H(\Omega) + H(C)]/2}. \tag{7}$$

Where I represents mutual information, as shown in equation (8).

$$I(\Omega, C) = \sum_k \sum_j \frac{|o_k \cap c_j|}{N} \log \frac{N|o_k \cap c_j|}{|o_k| \cdot |c_j|}. \tag{8}$$

H represents entropy, as shown in equation (9).

$$H(\Omega) = -\sum_i \frac{o_i}{N} \log \frac{o_i}{N}. \tag{9}$$

The value range of NMI is [0,1]. The closer the NMI is to 1, the better the clustering effect is.

The ARI makes up for the insufficient punishment of Rand coefficient RI, and its definition is shown in equations (10) and (11).

$$RI = \frac{a + b}{C_N^2}. \tag{10}$$

$$ARI = \frac{RI - mean(RI)}{max(RI) - mean(RI)}. \tag{11}$$

Where, a represents the logarithm of elements of the same class in both the actual class cluster and the cluster prediction class cluster. b represents different categories of actual class clusters, and is also the element logarithm of different categories in the cluster prediction cluster. $mean(RI)$ indicates the average value of RI. The value range of ARI is [0,1]. The closer the ARI is to 1, the better the clustering effect is.

4.4. Model Training

When $M_{encoder}$ and $M_{decoder}$ are spliced together, a twin neural network model M can be formed to calculate text similarity. The feature vectors h_i and h_j of the two texts t_i and t_j are obtained by $M_{encoder}$ respectively, and h_i and h_j are taken as the inputs of $M_{decoder}$, and the similarity P_c of the two texts is finally obtained.

In the process of clustering, the same text t_i needs to calculate the similarity with multiple texts. If the complete twinned neural network model M is used to calculate the similarity, a large time cost will be generated, and the same text t_i will be repeatedly encoded. We split M into $M_{encoder}$ and $M_{decoder}$, each text only needs to be encoded once by $M_{encoder}$, and the simple $M_{decoder}$ is used to calculate the similarity several times, which can greatly reduce the time cost of the clustering process.

During model training, we splice $M_{encoder}$ and $M_{decoder}$ into a twin neural network M, trained M on the training set for 5 rounds, and then split M into $M_{encoder}$ and $M_{decoder}$ to achieve model training. Model retraining is also to retrain the twin neural network M, the number of training rounds is 5 rounds, and then M is divided into $M_{encoder}$ and $M_{decoder}$, so as to achieve the model retraining.

The output of the neural network M is y_{pre}, and the true label is y_{true}, which is 0 or 1. The optimizer is Adam, and the cross entropy loss function is adopted. The calculation steps are shown in equation (12).

$$loss = -\frac{1}{N}\sum_{i=1}^{N} y_{true}\log(y_{pre}) + (1 - y_{true})\log(1 - y_{pre}). \tag{12}$$

4.5. Education Text Clustering Results

We compare the clustering results obtained by the proposed event clustering model with the following clustering methods.

Singlepass [32]: This clustering method uses vector space model to represent text, uses cosine to calculate text similarity, and uses singlepass algorithm for clustering, which belongs to unsupervised clustering algorithm.

K-means: This clustering method uses vector space model to represent the text, uses cosine to calculate the text similarity, and uses K-means algorithm to cluster, which belongs to unsupervised clustering algorithm.

LSH: This is a clustering algorithm based on local sensitive hashing, using a vector space model to represent the text, using cosine to calculate the text similarity, belonging to the unsupervised clustering algorithm.

Hadifar [33], the clustering model uses SIF Embedding to represent the text, autoencoder learning the low-dimensional feature representation of the text, and deep clustering algorithm is adopted for short text clustering, which is an unsupervised clustering algorithm. In this method, short texts from different fields are used as test sets in the experiment. In this experiment, the test sets are different events in the seismic field.

Wang [34]: This clustering model uses LSTM to extract text features, linear neural network model to calculate text similarity, and incremental clustering algorithm to cluster, which belongs to supervised clustering algorithm.

BERT: This method replaces the model of Wang et al. [34] with BERT word vectors and belongs to supervised clustering algorithm.

Since the clustering results are affected by the order of data input, we randomly scrambled the test data for 10 times of clustering, and recorded the average value of various clustering indicators. The clustering results are shown in Table 1.

Table 1. Cluster results

Model	Purity	NMI	ARI
Singlepass	0.65	0.67	0.50
K-means	0.87	0.86	0.76
LSH	0.59	0.48	0.31
Hadifar	0.61	0.56	0.45
Wang	0.89	0.90	0.71
BERT	0.74	0.71	0.58
Proposed	0.92	0.94	0.81

Among them, the supervised clustering model adopted by Wang [35] has better clustering effect than other clustering models, so this model is taken as the baseline. Under the same training, compared with the baseline model, the proposed model has improved in various clustering indexes, 3% in Purity, 4% in NMI and 10% in ARI.

The text in social media is short, the expression is arbitrary, and even the comments on the same event are expressed in different ways. The use of vector space model and word vector weighting to extract text features is easy to be affected by the interference words, so that the key feature information can not be prominent, resulting in poor clustering effect.

By training the model with supervision, the event features of the text can be identified more accurately and the clustering effect can be enhanced compared with the unsupervised clustering model. However, compared with Word2Vec word vector based clustering method, the clustering result has decreased, because BERT word vector model contains more information, and the model is easy to overfit existing events. The events in the test set did not appear in the training set, so the prediction effect is poor, and the BERT model takes more time than the Word2Vec word vector model, which is less efficient for massive social media texts.

The supervised clustering algorithm relies heavily on the training set. The key to make the model more accurately distinguish various events is whether there are sufficient event types in the training set. Therefore, 10, 15, 20, 25 and 30 education events are selected as the training data respectively, from which 350000 sets of texts are extracted to compose a training set and train the model. 20 education events in Section 3.1 are taken as the test set and the model of Wang et al. [35] is taken as the baseline. Taking NMI as a reference index, the test data is randomly disrupted for 10 times of clustering, and the average value of NMI is obtained, as shown in Figure 2.

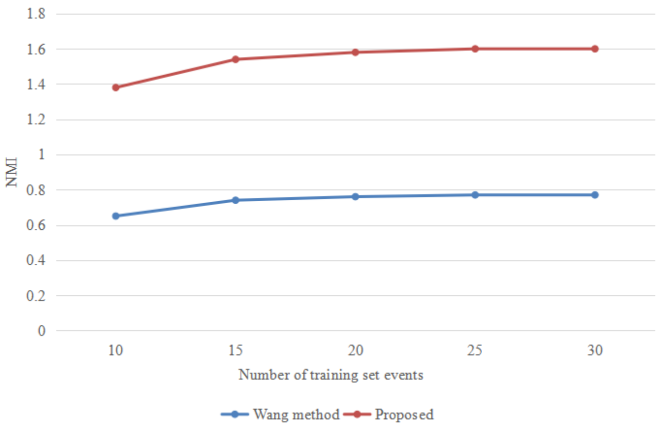

Fig. 2. The influence of the number of training set events on the clustering results

Both model retraining and uncertain sample reunion steps require setting some additional parameters. The update threshold U, which has the greatest influence on the clustering effect of new models, is mainly used to control the frequency of model retraining.

In this experiment, U is set as 10, 50, 200, 500 and 1000 respectively, and the sample number D of each training set is set as 4 times of U, which were 40, 200, 800, 2000 and 4000 respectively. With 10 educational events in Section 3.1 as the test set, Wang et al. [35] model as baseline, and NMI as the reference index, the test data are randomly disrupted for 10 times of clustering, and the NMI is averaged. The results obtained are shown in Figure 3.

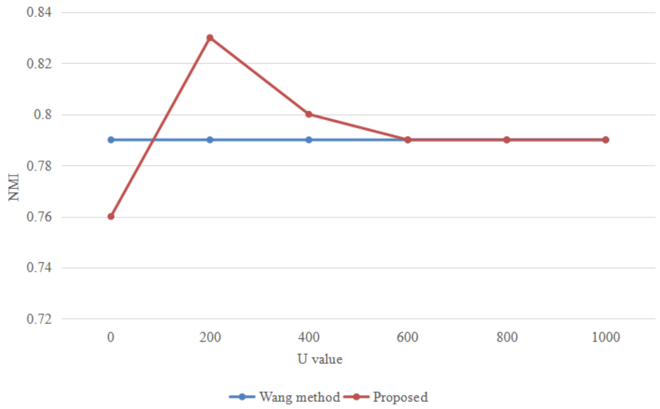

Fig. 3. Influence of U value on clustering results

As can be seen from the results of figure 3, when the U value is small, 10, the clustering effect is poor. The main reason is that the training set is small, and the probability of uneven distribution of categories is large. When U value gradually increases to 200, sample distribution tends to be uniform, and the clustering effect relative baseline will be significantly improved. However, increasing the U value will reduce the number of model retraining, resulting in a large number of data can only be clustered using the original model, so that the clustering results continue to approach but not lower than the baseline. Therefore, the value of U should be taken as small as possible under the condition that the data distribution is as even as possible, so that the proposed model can achieve the best clustering effect.

Table 2 and Figure 4 show the classification effects of different models.

Combined with the experimental results in Table 2 and Figure 4, the following conclusions can be drawn:

(1) It can be seen from the experimental results of K-means and LSH models that LSH has a slightly stronger ability to extract features than K-means, with an increase of 0.14% in accuracy and 0.55% in F1 value.

(2) Comparing experiments on Singlepass, Hadifar, LSH and Wang, it can be seen that BERT word embedding model is better than traditional word embedding to generate word vector. Compared with Singlepass model, the accuracy of Hadifar model is increased by 1.49%, and the F1 value is increased by 2.55%. Compared with the LSH model, the accuracy of the Wang model is increased by 1.37%, and the F1 value is increased by 2.31%.

Table 2. Cluster results

Model	Accuracy/%	F1/%
Singlepass	89.87	88.68
K-means	89.70	89.13
LSH	89.84	89.68
Hadifar	91.36	91.23
Wang	91.21	91.99
BERT	92.56	92.38
Proposed	93.88	93.74

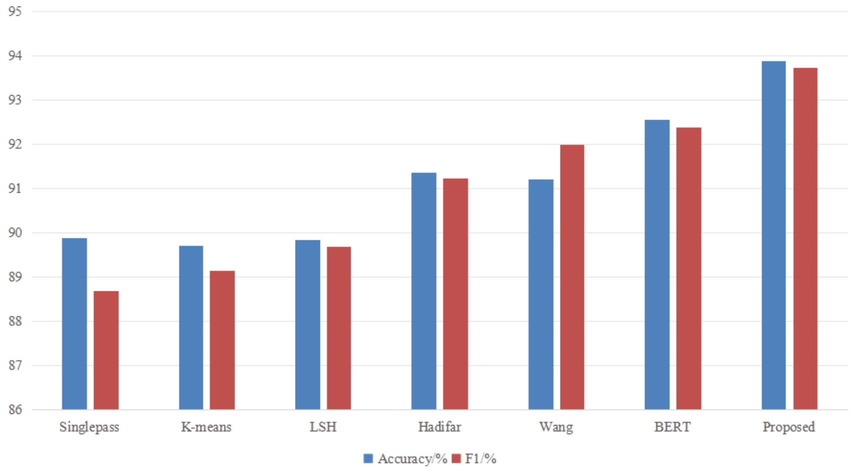

Fig. 4. Classification effect of different models

(3) According to the evaluation indicators of Hadifar, Wang, BERT and the Proposed model, it can be seen that comprehensive consideration of news headlines and text features is better than a general analysis of news text classification. Compared with the Hadifar model, the Proposed model has an increase of 2.52% in accuracy and 2.51% in F1 value. Compared with Wang model, the Proposed model is 2.67% higher in accuracy and 1.75% higher in F1 value.

After comprehensive analysis of the above experimental results, the advantages of the proposed model compared with other models mainly lie in that the proposed model comprehensively considers the differences between the features contained in the news headlines and the news text and fully exploits the text features of each. Since news headlines contain a large number of topic features and keyword features, this model uses LDA topic model to extract the topic features of headlines, Word2Vec model to extract the semantic features of headlines, and fuses the features to obtain the fusion features, and then uses TextCNN model to mine more fine-grained local features. For higher classification standards, just classifying news headlines is far from enough. Due to the short length of news headlines, they lack certain contextual information to supplement the theme features and keyword information contained in them, which easily leads to classification only according to the literal meaning of the headlines. Therefore, this model extracts the context information from the news text on the basis of the classification of news headlines. In the classification of news text, this paper uses BERT model to obtain text word vectors containing context information and complex semantics, and uses BERT to further mine the context information, so as to supplement the topic information and keyword information contained in news headlines. Therefore, the classification model in this paper has been improved compared with other models in various indexes.

The specific classification results of the model proposed in this paper in 10 categories of news are shown in Table 3 and Figure 5. The performance indicator is accuracy.

Table 3. Classification effect of different categories with proposed method

Class	Number	Precision/%	Recall/%	F1/%
Finance	0	91.97	92.80	92.38
property	1	95.39	95.20	95.29
stock	2	89.49	88.60	89.04
education	3	97.38	96.80	97.08
technology	4	89.76	90.30	90.02
society	5	92.58	93.60	93.08
politics	6	90.99	92.01	91.49
sports	7	98.69	98.20	98.44
game	8	96.66	95.10	95.23
recreation	9	95.38	95.10	95.23

From the above results, it can be seen that the model in this paper has a good effect on the classification of news texts in the fields of sports, education and games, especially in the field of sports, with its accuracy rate 98.69%, recall 98.20%, and F1 98.44%. The worst results are in the stock and technology sector, where the accuracy rate of the stock

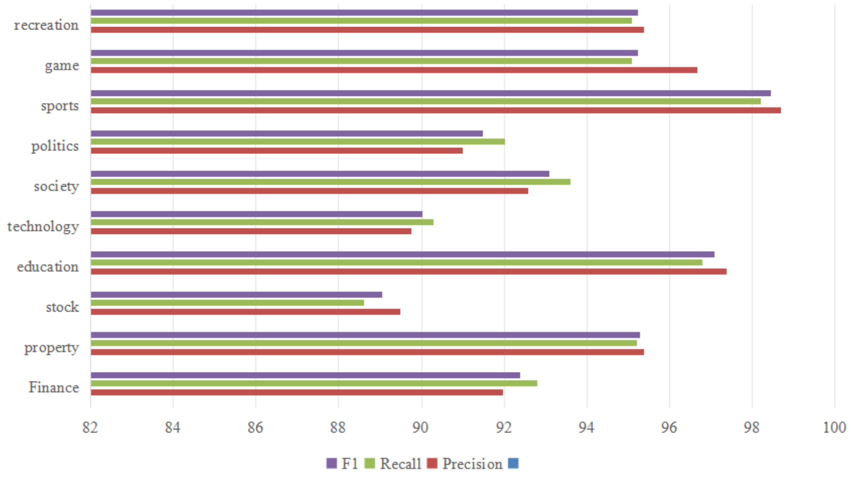

Fig. 5. Classification effect of different categories with proposed method

sector is 89.49%, the recall rate is 88.60%, and the F1 value is 89.04%; The accuracy rate in the technology sector is 89.76%, the recall rate is 90.30%, and the F1 value is 90.02%.

5. Conclusions

Efficient ubiquitous learning can achieve accurate service. Once learners have the right equipment and network, they can get the right resources pushed by the system accurately in the right time, and improve the effective use of information resources. Once the knowledge content gets the rapid flow, its utilization value will be maximized. In the process of research, it is necessary to explore how to optimize and maximize the value of resources in the process of dissemination and use, so as to promote the healthy and sustainable development of information resources.

This paper proposes a novel English social education text event clustering model, which uses DFCEDN to extract text features and linear model to calculate text similarity for incremental clustering. The samples assigned with uncertainty in the clustering process are reclustered after the end. The retraining process can make the model learn new event information, so that the accuracy of the model is constantly improved with the clustering process. Reclustering of uncertain samples can prevent uncertain samples from affecting the centroid representation, reduce the probability of retraining the model with wrong samples, and improve the clustering accuracy of uncertain samples. Compared with the supervised cluster model with the same pre-training, the clustering index of the proposed model is improved.

Acknowledgments. Authors are grateful for the anonymous review by the review experts.

References

1. Cinelli M, De Francisci Morales G, Galeazzi A, et al. The echo chamber effect on social media[J]. Proceedings of the National Academy of Sciences, 2021, 118(9): e2023301118.
2. 2 Li F, Larimo J, Leonidou L C. Social media marketing strategy: definition, conceptualization, taxonomy, validation, and future agenda[J]. Journal of the Academy of Marketing Science, 2021, 49: 51-70.
3. 3 Kross E, Verduyn P, Sheppes G, et al. Social media and well-being: Pitfalls, progress, and next steps[J]. Trends in Cognitive Sciences, 2021, 25(1): 55-66.
4. 4 *Ö*rs F K, Yeniterzi S, Yeniterzi R. Event clustering within news articles[C]//Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020. 2020: 63-68.
5. 5 Rybski D, Buldyrev S V, Havlin S, et al. Communication activity in a social network: relation between long-term correlations and inter-event clustering[J]. Scientific reports, 2012, 2(1): 1-8.
6. 6 Xu J, Xu B, Wang P, et al. Self-taught convolutional neural networks for short text clustering[J]. Neural Networks, 2017, 88: 22-31.
7. 7 Costache R, Pham Q B, Arabameri A, et al. Flash-flood propagation susceptibility estimation using weights of evidence and their novel ensembles with multicriteria decision making and machine learning[J]. Geocarto International, 2021: 1-33.
8. 8 Balogun A O, Basri S, Capretz L F, et al. An adaptive rank aggregation-based ensemble multi-filter feature selection method in software defect prediction[J]. Entropy, 2021, 23(10): 1274.
9. 9 Long M, Cao Z, Wang J, et al. Conditional adversarial domain adaptation[J]. Advances in neural information processing systems, 2018, 31.
10. 10 Akritidis L, Alamaniotis M, Fevgas A, et al. Confronting sparseness and high dimensionality in short text clustering via feature vector projections[C]//2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2020: 813-820.
11. 11 Ma W, Hu X, Chen C, et al. Social media event prediction using DNN with feedback mechanism[J]. ACM Transactions on Management Information Systems (TMIS), 2022, 13(3): 1-24.
12. 12 Hompes B, Buijs J C A M, van der Aalst W M P, et al. Detecting Change in Processes Using Comparative Trace Clustering[J]. SIMPDA, 2015, 2015: 95-108.
13. 13 Adamopoulos P, Ghose A, Todri V. The impact of user personality traits on word of mouth: Text-mining social media platforms[J]. Information Systems Research, 2018, 29(3): 612-640.
14. 14 Liu J, Gao L, Guo S, et al. A hybrid deep-learning approach for complex biochemical named entity recognition[J]. Knowledge-Based Systems, 2021, 221: 106958.
15. 15 Yang Z, Yao F, Fan K, et al. Text dimensionality reduction with mutual information preserving mapping[J]. Chinese Journal of Electronics, 2017, 26(5): 919-925.
16. 16 Rezaee M J, Eshkevari M, Saberi M, et al. GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game[J]. Knowledge-Based Systems, 2021, 213: 106672.
17. 17 Lilleberg J, Zhu Y, Zhang Y. Support vector machines and word2vec for text classification with semantic features[C]//2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC). IEEE, 2015: 136-140.
18. 18 Vodrahalli K, Chen P H, Liang Y, et al. Mapping between fMRI responses to movies and their natural language annotations[J]. NeuroImage, 2018, 180: 223-231.
19. 19 Soni S, Chouhan S S, Rathore S S. TextConvoNet: a convolutional neural network based architecture for text classification[J]. Applied Intelligence, 2022: 1-20.
20. 20 Liu D, Shan L, Wang L, et al. P3oi-melsh: Privacy protection target point of interest recommendation algorithm based on multi-exploring locality sensitive hashing[J]. Frontiers in Neurorobotics, 2021, 15: 660304.

21. 21 Zhao Z, Gao M, Luo F, et al. LSHWE: improving similarity-based word embedding with locality sensitive hashing for cyberbullying detection[C]//2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020: 1-8.
22. 22 Batanović V, Petrović M M. Cross-Level Semantic Similarity for Serbian Newswire Texts[C]//Proceedings of the Thirteenth Language Resources and Evaluation Conference. 2022: 1691-1699.
23. 23 Enguehard J, OHalloran P, Gholipour A. Semi-supervised learning with deep embedded clustering for image classification and segmentation[J]. IEEE Access, 2019, 7: 11093-11104.
24. 24 Zhao M, Wang M, Ma Y, et al. CEIL: A General Classification-Enhanced Iterative Learning Framework for Text Clustering[C]//Proceedings of the ACM Web Conference 2023. 2023: 1784-1792.
25. 25 Pavlinek M, Podgorelec V. Text classification method based on self-training and LDA topic models[J]. Expert Systems with Applications, 2017, 80: 83-93.
26. 26 Zhang X, Zhang L. Topics extraction in incremental short texts based on LSTM[J]. Social Network Analysis and Mining, 2020, 10(1): 83.
27. 27 Teng L, Qiao Y, Shafiq M, et al. FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction[J]. IEEE Transactions on Network and Service Management, 2023. DOI: 10.1109/TNSM.2023.3273991
28. 28 Li P, Laghari A A, Rashid M, et al. A deep multimodal adversarial cycle-consistent network for smart enterprise system[J]. IEEE Transactions on Industrial Informatics, 2022, 19(1): 693-702.
29. 29 Liu Y, Tong D, Liu X. Measuring spatial autocorrelation of vectors[J]. Geographical Analysis, 2015, 47(3): 300-319.
30. 30 Iscen A, Furon T, Gripon V, et al. Memory vectors for similarity search in high-dimensional spaces[J]. IEEE transactions on big data, 2017, 4(1): 65-77.
31. 31 Yin S, Li H, Sun Y, et al. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey[J]. IJLAI Transactions on Science and Engineering, 2024, 2(2): 13-20.
32. 32 Marroig G, Cheverud J. Size as a line of least resistance II: direct selection on size or correlated response due to constraints[J]. Evolution, 2010, 64(5): 1470-1488.
33. 33 Forestiero A, Pizzuti C, Spezzano G. A single pass algorithm for clustering evolving data streams based on swarm intelligence[J]. Data Mining and Knowledge Discovery, 2013, 26: 1-26.
34. 34 Hadifar A, Sterckx L, Demeester T, et al. A self-training approach for short text clustering[C]//Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019). 2019: 194-199.
35. 35 Wang H, Li F. A text classification method based on LSTM and graph attention network[J]. Connection Science, 2022, 34(1): 2466-2480.

Zhenping Jing is with the School of Foreign Languages, Zhengzhou University of Science and Technology. Research interests are Cluster analysis, English semantic analysis, Big Data research, Artificial intelligence.

Received: April 18, 2024; Accepted: June 29, 2024.