Journal of Network Intelligence ©2026 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **11**, Number **1**, February **2026**

Research on Business English Text Clustering Based on Improved DBSCAN Algorithm

Yun Long[1,*]

[1] Shunde Polytechnic University, Foshan 528300, P. R. China
18923170729@163.com

Li-Xin Chen[2]

[2] University of Central Lancashire, Preston, Preston PR1 2HE, UK
LChen30@uclan.ac.uk

*Corresponding author: Yun Long

Received November 20, 2024, revised February 18, 2025, accepted May 27, 2025.

Abstract. *As the science and technology rapidly growing, business English produces exponential growth of text data, how to analyze its clustering is of great significance. This article offers a business English text clustering method relied on the optimized Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to address the issues of existing study on the sensitivity of noisy data and the single shape of clustering. The DBSCAN algorithm is first optimized using nearest-neighbor assignment (EDBSCAN) to assign unassigned points to the most probable clusters, making full use of the information of the assigned clusters within the nearest-neighbor to avoid chaining errors. Then the Skip-gram model is introduced to learn the text word vector representation, and the TF-IDF algorithm is optimized by inter-class and intra-class discretization to achieve efficient extraction of text features. Finally, the obtained text feature words are subjected to a weighting operation, and by calculating the feature similarity between the text feature words, it is judged whether the threshold condition is satisfied or not, and the text that satisfies the threshold is subjected to EDBSCAN clustering, which clusters the similar text into one class. The experimental outcome indicates that the clustering accuracy of the offered method is 91.2% and the Adjusted Mutual Information (AMI) is 0.4246, which verifies that the clustering effect of the suggested approach is better than other methods.*

Keywords: DBSCAN algorithm; Text clustering; Nearest neighbor assignment; Skip-gram model; TF-IDF algorithm.

1. **Introduction.** As one of the important branches of English for Specialized Purposes, Business English is an important foundation and guarantee for the development of foreign communication activities, and its important position in related activities cannot be ignored [1]. As the science and technology growing, the amount of business English text data is growing exponentially [2]. Therefore, in the huge amount of data, it is very important to analyze these data and mine the hidden information in them. Data mining technology can mine valuable information from massive data for all walks of life and bring great economic benefits to people [3]. Cluster analysis is an essential tool commonly used in data mining with the aim of dividing a dataset consisting of obtained information into multiple clusters consisting of similar objects, with a high degree of variability between

different clusters [4]. Thus, how to efficiently use cluster analysis methods to mine useful information from massive business texts has become an urgent problem.

1.1. **Related work.** The study of business English text clustering belongs to the field of text clustering. Naeem and Wumaier [5] combined the bee colony optimization algorithm and introduced the "fair operation" and "clone operation" for refining the K-means center of gravity, which was finally applied to the text clustering of English text clustering. Ahmed et al. [6] reconstructed the text clustering center iterative formula and measurement function to improve K-means by maximum distance selection of centers method. Suryanarayana et al. [7] found that optimizing K-modes using genetic algorithms produces better clustering outcome for high dimensional data sets encountered in the field of clustering English text. Das et al. [8] used an improved artificial bee colony algorithm to select suitable clustering centers, and merged with the conventional K-means approach for text clustering, but there are still problems such as local optimal solutions. In addition to the K-means, K-modes algorithm, some scholars have combined the nearest neighbor propagation algorithm and hierarchical clustering, using the idea of "divide and conquer", firstly dispersing the dataset, and then combining the text clustering results [9, 10]. Ran et al. [11] proposed a hierarchical based BIRCH method for text clustering, but the time complexity is large and it is not possible to make human interventions and corrections in the process of clustering. Lang et al. [12] step-by-step optimize BIRCH in terms of the distribution of feature terms within classes, between classes, and across different texts within classes, respectively. Soares et al. [13] proposed a hierarchical text clustering algorithm with self-learning ability by combining word frequency weights and cosine similarity to enhance the clustering performance by updating the category keywords, but the clustering results are not stable.

Density-based clustering algorithms divide class clusters according to the density of data points, do not require to specify the amount of clusters in advance, can recognize class clusters of random shapes, and is with a strong capability to deal with noisy data. The main representative algorithm is Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [14], which not only improves the speed and accuracy of text clustering, but also enhances the stability and reliability of the algorithm in coping with diverse data sets [15]. Yin et al. [16] introduced an improved idea of determining the truncation distance and clustering centers based on the approximate distance from the point of maximum density to the point of minimum density as well as the change of similarity between the points that may become the clustering centers, which realized the optimization of DBSCAN, but the accuracy of clustering is not high. Abualigah et al. [17] used the least squares method to reduce the high dimensional data to low dimensions and changed the eps value in the DBSCAN algorithm to self-adjustment, and the text clustering results were better. Sangaiah et al. [18] used the TF-IDF method for feature word weight calculation, while in the DBSCAN algorithm to prioritize the selection of global high-quality points as the kernel, and then gradually confirm whether there is a conflict with other kernels, but the final clustering results show a chain. Al-Betar et al. [19] used a document vector space model (VSM) based on neighborhood ontology to adjust the feature word weights with semantic relations and used dynamic solving on DBSCAN algorithm to get the optimal eps value, and the improved algorithm can find the topics with higher relevance faster.

1.2. **Contribution.** Due to the advantages of DBSCAN algorithm which is insensitive to noisy data and can divide the data into different clusters, it has received wide attention from researchers. In this article, the defects of DBSCAN algorithm are optimized and the improved DBSCAN algorithm (EDBSCAN) is applied to business English text clustering.

1. Optimize DBSCAN using weighted nearest neighbor allocation, which constantly updates the data state and fully considers the correlation between data to avoid "chain error". By making full use of the information of the assigned clusters in the nearest neighbors, the clustering results will not be affected even if there is an error point, so as to improve the robustness of the algorithm.
2. The Skip-gram model is introduced to represent the word vectors of Business English text, and the TF-IDF does not take into account the impact of the distribution information of characteristic items between and within classes on the calculation of the weights, and the TF-IDF method is modified by utilizing the interclass discretization degree and the intraclass discretization degree, so as to realize the efficient capturing of text features.
3. The similarity of text feature words is calculated based on the Hamming distance, and the text with similarity lower than the threshold is treated as isolated points. The EDBSCAN algorithm is utilized to cluster the text, and the similar texts are clustered into one class to realize the efficient management of the text.
4. To estimate the clustering effect of the offered approach, taking the business English text data of an enterprise as the experimental dataset, and taking the accuracy rate, NMI, ARI, etc. as the evaluation indexes and discussing the clustering results, the experiment verifies the effectiveness of the EDBSCAN algorithm in business English text clustering.

2. Theoretical analysis.

2.1. Text clustering process.

The biggest difference between text clustering as an unsupervised learning and classification is that clustering does not need to be trained and learned based on the features and attributes of the data. An efficient text clustering method can finally process such messy and unstructured data as text into a data set with high similarity within clusters, which is convenient for users to collect text information [20]. The approximate process of text clustering is implied in Figure 1. The process of text clustering consists of text preprocessing, invoking clustering algorithms and result evaluation.

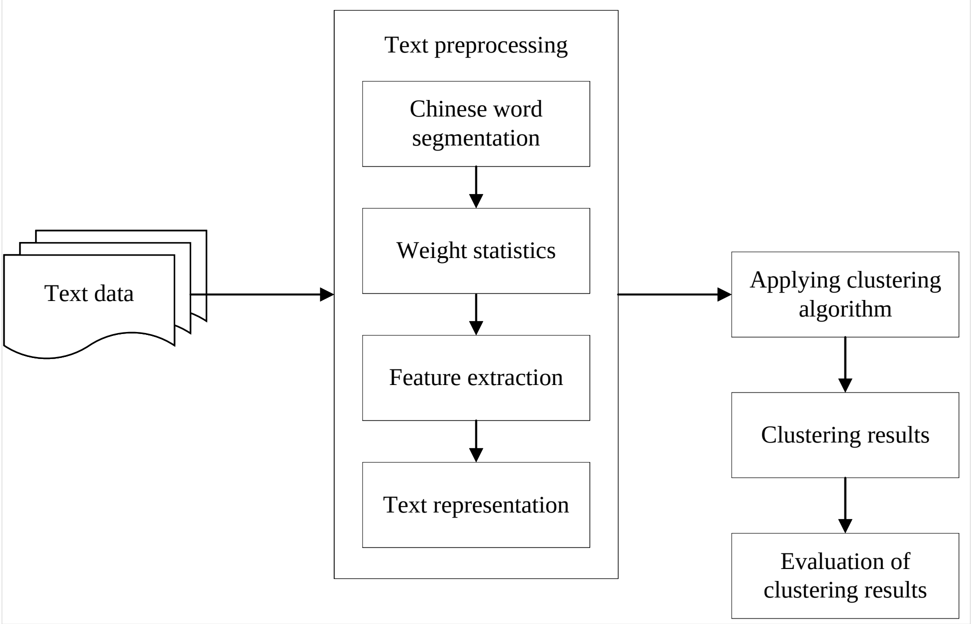

Figure 1. The process of text clustering

2.2. **TF-IDF algorithm.** TF-IDF is a usually adopted text feature extraction approach, the fundamental idea is that the more frequently a feature word occurs in a text, the more important it is. TF-IDF consists of word frequency (TF) and inverse document frequency (IDF) [21]. Let the text be d, and d consists of T_d feature words, then the amount of times the k-th feature word t_k appears in d is T_i, then the word frequency of t_k is as follows.

$$TF_{t_k} = \frac{T_i}{T_d} \tag{1}$$

The second part of the inverse document frequency is the total amount of words in the text dataset divided by the amount of texts containing the featured word, and then take the logarithm of the above result. Let there are a total of D_d texts in the text data set, where t_k appears in D_t texts, then the inverse document frequency of t_k is as follows.

$$IDF_{t_k} = \log\left(\frac{D_d}{D_t}\right) \tag{2}$$

Finally, the TF-IDF for the k-th t_k in text d is computed as follows.

$$TF - IDF_{t_k} = TF_{t_k} * IDF_{t_k} \tag{3}$$

Therefore, TF-IDF prefers to eliminate the words that occur many times in the text dataset, so that the feature words have more obvious words.

2.3. **DBSCAN clustering algorithm.** The DBSACN approach is a classical representative of density-based clustering approaches that can handle not only spherical clusters, but also non-convex data and clusters of arbitrary shapes found [22]. Compared to clustering approaches for example K-means and K-modes, DBSCAN is able to deal with noisy datasets, and can be adjusted by parameters to determine the degree of noise tolerance, with strong stability. The important concepts of the DBSCAN algorithm are as follows, and a schematic of each concept is shown in Figure 2.

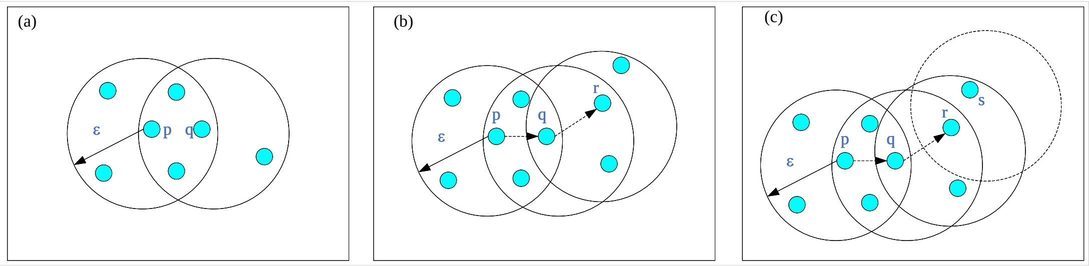

Figure 2. Schematic conceptualization of the DBSCAN algorithm: (a) direct densities up to; (b) density up to; (c) density connection

1. Neighborhood ξ: Given an object p in space, a circular region with p as the center and ξ as the radius.
2. Density: the number of all data points in the ξ-neighborhood of data point p.
3. Core Objects (Core Points): Given $MinPts$ (Minimum Density Reachable Points), an object is a core object if it contains at least $MinPts$ of objects in its ξ-neighborhood.
4. Boundary point: If a non-core object is in the ξ-neighborhood of a core object, the core object is said to be a boundary point.
5. Density Reachability: If p, q, r are core objects, q is a direct density reachable point of p, and object r is a direct density reachable point of q, then r is a density reachable object of p in terms of ξ and $MinPts$.

6. Density connected: if q, r are core objects and s is a non-core object, q and s are reachable from the density of r with respect to ξ and $MinPts$, then s is density connected to q.

The algorithm, for any data x_i, only needs to compute the local density ρ_i and the relative distance δ_i, which are only related to the distance between x_i. ρ_i and δ_i define as follows.

$$\rho_i = \sum_j \chi(d(x_i, x_j) - d_c) \tag{4}$$

$$\delta_i = \begin{cases} \min_{j:\rho_i<\rho_j}\{d(x_i, x_j)\}, & if\ \exists\rho_i < \rho_j \\ \max_j\{d(x_i, x_j)\}, & otherwise \end{cases} \tag{5}$$

where $d(x_i, x_j)$ is the Euclidean distance between samples x_i and x_j, d_c is the cutoff distance, specified by the user. The function $\chi(x)$ is defined as follows

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \tag{6}$$

3. **DBSCAN algorithm based on nearest neighbor allocation optimization.** The DBSCAN algorithm is widely used due to its capability to handle clusters of random shapes, automatically discover the number of clusters and recognize noise. However, after determining the cluster centers, the allocation strategy of this algorithm may lead to error propagation and affect the clustering results. For this reason, this article adopts the idea of nearest neighbor assignment to optimize the DBSCAN algorithm (EDBSCAN), which uses the obtained clustering information to calculate the probability of the sample points, and assigns the unassigned points to the most probable clusters in order to avoid the "chain error".

The EDBSCAN algorithm first calculates the local gap density and distinguishes the core and boundary points according to the threshold, then removes the cross edges and forms the cluster backbone according to the maximum vertex base, and finally assigns the remaining points to the formed cluster backbone using the weighted K-nearest neighbor assignment method to form the final cluster.

1. Calculate the local gap density. For a given data point $X = \{x_1, x_2, \ldots, x_n\}$, construct a K-neighborhood graph whose adjacency matrix $W = [w_{i,j}]$ is defined as follows.

$$W_{i,j} = \begin{cases} d_{i,j}, & x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i) \\ 0, & otherwise \end{cases} \tag{7}$$

where $d_{i,j}$ is the Euclidean distance between a pair of points (x_i, x_j), $KNN(x_i)$ is the set of points in the K-neighborhood of point x_i, $k = p * n$, p is the percentage of the input sample, and n is the amount of samples.

With the density calculation in DBSCAN, it is known that points with high local density have a lot of points linked to them in the K-neighborhood picture. Therefore, to better compute the local denseness of the data, the local denseness point should meet an extra condition: the average weight of the edges connected to the point is small in the K-neighborhood domain picture, and the local denseness of x_i is now defined as follows.

$$\rho_i = \frac{|L_i^k|}{A_i^k} \tag{8}$$

where L_i^k is the set of all points linked to point x_i in the K-neighborhood picture, and $A_i^k = \sum_{x_j \in L_i^k} W_{i,j}/|L_i^k|$ denotes the average weight of edges connected to x_i in the K-neighborhood graph, so ρ_i can be recalculated as follows.

$$\rho_i = \frac{|L_i^k|^2}{\sum_{x_j \in L_i^k} W_{i,j}} \tag{9}$$

Since the densities of different datasets are different, in order to better obtain their densities, the above local densities are normalized and defined as local gap densities.

$$g_i = \frac{\rho_i}{\max\{\rho_m | x_m \in L_i^k\}} \tag{10}$$

2. Delete the crossing edges to form the cluster backbone. Set τ as the threshold to distinguish between core and boundary points, if $g_i \geq \tau$, then x_i is an essential point, and vice versa, the point is a boundary point, denoted by B. In a K-neighborhood graph, there may be some crossed cluster edges, and the following definition is made to better remove these crossed clusters. Let $l_{i,j}$ be an edge of x_i and x_j in the K-neighborhood graph, and if $l_{i,j}$ satisfies $L_i^k \cap B \neq \varnothing$, $L_j^k \cap B \neq \varnothing$, and $w_{i,j} \geq \max\{w_{i,u}, w_{v,j}\}$, then it is a potential cross-over edge, which is removed. Deleting the crossing edges results in a new graph in which all points in the first c components with maximum vertex bases are assigned to c subclusters, where c is the amount of clusters.
3. K-nearest neighbor assignment strategy. After determining the initial c sub-clusters, the K-nearest neighbor approach is used to assign the unassigned points to the most likely clusters. First, calculate the probability p_i^c of each point belonging to each of the c clusters, and find the largest probability p_i^c. If $p_i^c > 0$, assign the point to the cluster with the largest probability, and then re-calculate the point and loop the above steps, and if $p_i^c = 0$, increase the value of K by 1, and then continue to calculate and assign the point until all the points have been assigned.

$$p_i^c = e^{-\sum_{j \in KNN(x_i), y_j = c} \lambda_{i,j} * w_{i,j}} \tag{11}$$

where $\lambda_{i,j}$ is the weight as shown below.

$$\lambda_{i,j} = \frac{w_{i,j}}{\sum_{j \in KNN(x_i)} w_{i,j}} \tag{12}$$

With the above assignment strategy, the latest state of the data is used to assign labels, and the robustness of the algorithm is greatly improved by taking into account the effect of the occurrence of error points on the clustering results of the "chain error", so that even if there is an isolated error point, the labeling of that point will not be affected.

4. A Study on business English text clustering based on improved DBSCAN algorithm.

4.1. Word vector representation of business English text based on Skip-gram modeling.

To improve the clustering effect of business English text, this paper firstly utilizes Skip-gram model to learn the distributed representation of text words. Then the inter-class and intra-class discretization are introduced to modify the TF-IDF method to realize the efficient extraction of text features. By calculating the feature similarity between text feature words and judging whether it meets the threshold condition, the text that meets the threshold is subjected to EDBSCAN clustering, which makes full use of the information of the assigned clusters within the nearest neighbors and improves the robustness of the algorithm. The clustering process of the offered method is shown in Figure 3.

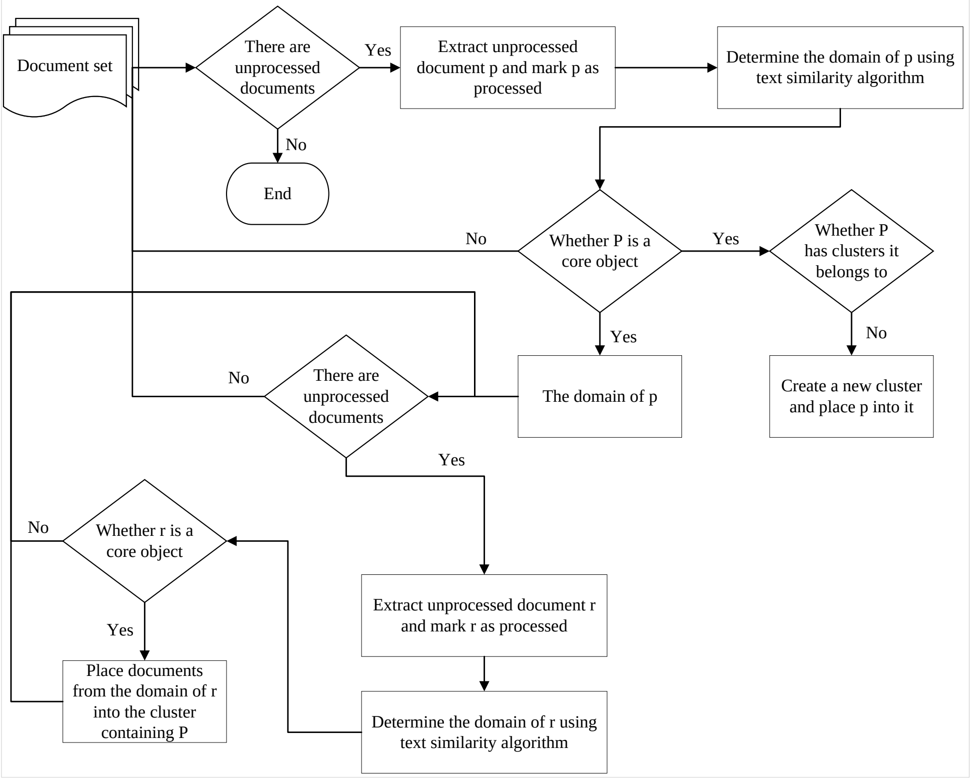
Document set
There are unprocessed documents
Yes
No
End
Extract unprocessed document p and mark p as processed
Determine the domain of p using text similarity algorithm
Whether P is a core object
No
Yes
Whether P has clusters it belongs to
No
Create a new cluster and place p into it
The domain of p
There are unprocessed documents
No
Yes
Extract unprocessed document r and mark r as processed
Determine the domain of r using text similarity algorithm
Whether r is a core object
No
Yes
Place documents from the domain of r into the cluster containing P

Figure 3. The clustering process of the offered method

Business English texts are natural language descriptions that are easily understood by humans, and it is difficult for computers to process their semantics directly. Therefore, in order to cluster the text, it is necessary to represent the business English text in a data format that can be recognized and processed by computers. Classical text representation models include Vector Space Model (VSM) [23], CBOW [24], Skip-gram [25], in which the algorithmic idea of Skip-gram model predicting the context with the help of a central word makes its input layer contain only the word vector of a central word, which not only improves the speed of the model training, but also improves the effect of the word vector.

Positive samples are constructed for the center word of the text by selecting words from the context within an appropriately sized window, with each positive sample corresponding to k negative samples. Negative samples can be generated randomly from the dictionary, given a central word w_t, any word w_i from the dictionary can be selected according to a certain probability distribution to form a pair of negative samples (W_t, W_i). The probability density function used is as below.

$$p(w_i) = \frac{f(w_i)^{\frac{3}{4}}}{\sum_{j=1}^{n} f(w_j)^{\frac{3}{4}}} \tag{13}$$

where $f(w_i)$ is the frequency of occurrence of word w_i in the dataset and n is the total amount of words in the corresponding dictionary.

Given a central word w and the corresponding context $Content(w)$, the expectation is to maximize the following target operation.

$$g(w) = \prod_{w' \in Content(w)} \prod_{u \in \{w'\} \cup NEG(w')} p(u|\ w') \tag{14}$$

where $NEG(w)$ is the set of negative instances corresponding to the center word w. $p(u|\ w')$ is computed as follows.

$$p(u|\ w') = \left(\sigma(v(w')^T\theta^u)\right)^{L^{w'}(u)} \cdot \left(1-\sigma(v(w')^T\theta^u)\right)^{1-L^{w'}(u)} \tag{15}$$

For the whole business English text corpus E can be obtained $G = \prod_{w\in E} g(w)$, which is logarithmic and inverted, and the final loss function can be obtained by substituting into Equation (14) as follows.

$$L = -\sum_{w\in E} \log \prod_{w'\in Content(w)} \prod_{u\in\{w'\}\cup NEG(w')} p(u|\ w') \tag{16}$$

The final iterative solution of Equation (16) using the gradient descent method yields the word vector d corresponding to each word in E. The word vectors are then solved using the gradient descent method.

4.2. **Business English text feature extraction based on improved TF-IDF approach.** Word vector representation of business English text will get a collection containing a huge number of words, and if these texts are clustered directly, a high-dimensional feature vector will be obtained, which will affect the clustering accuracy. To deal with the above issue, text features need to be extracted. The calculation process of TF-IDF is relatively simple, easy to implement and understand, and it is a commonly used text feature extraction method, but it does not take into account the influence of the distribution information of the feature items between and within classes on the calculation of the weights, so this paper utilizes the interclass discretization and intraclass discretization to modify the TF-IDF method, so as to carry out the efficient extraction of text features.

1. The inter-class scatter of the feature term D_{ac}. D_{ac} describes the degree of equalization of the distribution of the feature term t in each class and is calculated as follows.

$$D_{ac} = \frac{\sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(tf_i(t) - \overline{tf(t)})^2}}{\overline{tf(t)}} \tag{17}$$

   where m is the amount of categories, $tf_i(t)$ is the frequency of occurrence of t in class i, $\overline{tf(t)}$ is the average of the frequency of occurrence of t in each category, and $tf(t)$ is the total frequency of occurrence of t in each category.

2. The within-class dispersion of the characteristic term D_{ic}. D_{ic} is the degree of equilibrium of the distribution of t in a class and is calculated as follows.

$$D_{ic} = \frac{\sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(tf'_j(t) - \overline{tf'(t)})^2}}{\overline{tf'(t)}} \tag{18}$$

   where n is the total number of documents in the class, $tf'_j(t)$ is the frequency of occurrence of t in the j-th document, $\overline{tf'(t)}$ is the average of the frequency of occurrence of t in each document, and $|tf'_j(t)|$ is the total frequency of occurrence of t in each document.

Combining the above two considerations, together with the Sigmoid function processing of the traditional TF-IDF formula, the following improved TF-IDF formula is obtained.

$$f = w_i(d_j) = \frac{D_{ac}\times(1-D_{ic})}{(1+e^{-(TF_{ij}*IDF_i)})} \tag{19}$$

where f is the text feature, TF_{ij} is the word frequency $T_{t_{ij}}/T_d$ of t, d is the text, IDF_i is the inverse document frequency $\log(N_d/N_{t_i})$, N_d is the amount of texts, $T_{t_{ij}}$ is the amount of occurrences of t, and T_d is the amount of times t occurs.

4.3. **Business English text clustering based on improved DBSCAN algorithm.** After the weighting operation on the set of business English text feature vocabulary using the TF-IDF method, a feature matrix f is obtained. The text similarity approach is adopted to compute the similarity between document d and other documents in the document set, if the similarity between other documents and d is greater than or equal to a set threshold, the EDBSCAN algorithm will store the documents that satisfy the threshold condition in the neighborhood A of d. If d does not belong to a cluster, a new cluster will be built and d will be put into this cluster, and the algorithm will end until all the text has been processed.

The traditional text similarity computation is relied on the Euclidean distance, but the text similarity computation method relied on the Hamming distance is much easier to operate [26]. Before the text is encoded in Hamming, it is first necessary to arrange the characteristics of the text into an n-bit sequence of code words, the information in the text is expressed in these code words, and a one-to-one mapping relationship is established between the text and the code words. Let the codeword of text d_1 be $M_1 = (x_1, x_2, \ldots, x_n)$ and the codeword of text d_2 be $M_2 = (y_1, y_2, \ldots, y_n)$, then the similarity formula based on Hamming distance between d_1 and d_2 is as follows.

$$sim(M_1, M_2) = 1 - \left(\sum_{i=1}^{n} x_i \oplus y_i \right) / n \tag{20}$$

Based on the above text similarity calculation, this paper utilizes EDBSCAN to cluster the business English text, the steps in detail are as bellow.

1. Construct the original eigenvector matrix f, f consists of n m-dimensional random vectors, as shown in Equation (21). Perform normalization transformation on f to obtain the transformed matrix z_{ij}, as shown in Equation (22).

$$f = \begin{bmatrix} f_1^T \\ f_2^T \\ \vdots \\ f_n^T \end{bmatrix} = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,m} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n,1} & f_{n,2} & \cdots & f_{n,m} \end{bmatrix} \tag{21}$$

$$z_{ij} = \frac{f_{ij} - \overline{f_j}}{s_j} \tag{22}$$

where $\overline{f_j} = \sum_{i=1}^{n} f_{ij}/n$ and $s_j = \frac{1}{n}(|f_{1j} - \overline{f_j}| + |f_{2j} - \overline{f_j}| + \cdots + |f_{nj} - \overline{f_j}|)$ are the mean and the absolute deviation from the mean of the j-th vector, respectively.

2. The transformation of the normalized data matrix into a kernel function matrix using the kernel function is equivalent to mapping the input data into a high-dimensional characteristic space F via a nonlinear operation, where $F = \{\Phi(z_1), \Phi(z_2), \ldots, \Phi(z_n)\}$, the elements of the kernel function matrix are as follows.

$$K_{ij} = K(z_i, z_j) \tag{23}$$

3. Through the kernel function to set the neighborhood eps, using the Equation (20) to calculate the similarity between document d and other documents in the document set, if the similarity with d is greater than or equal to the set threshold, it will be stored in the d eps.
4. If the similarity is less than the set threshold, scan the text dataset, find out the min and max values of each dimensional data to calculate the size of each dimensional interval and establish a hash table, select any one of the unprocessed data object p, query its ξ neighborhood $N_\xi(p)$ by hash, if $N_\xi(p)$ contains objects not less than $MinPts$, establish a novel cluster C, and add all the points in $N_\xi(p)$ to C.

5. Use Equation (24) to compute the $2m$ objects in $N_\xi(p)$ that are furthest away from p and extend the class as representative-seeds objects.

$$d(p_1, p_2) = \sqrt{K(p_1, p_1) - 2K(p_1, p_2) + K(p_2, p_2)} \tag{24}$$

6. For each representative object q, if q is a core object, add $N_\xi(q)$ to C. Otherwise, compute the probability that all points belong to each of the m clusters p_i^m, and allocate objects by determining the cluster where the maximum probability is located.
7. If q is not a core object, but the objects contained within $N_\xi(q)$ are greater than or equal to $0.8MinPts$, then the unclassified objects in $N_\xi(q)$ are added to C.
8. Repeat steps (1) through (7) until all representative objects have been processed.

5. Experiment and result analysis.

5.1. **Clustering accuracy analysis.** Based on the business English text data of an enterprise, this paper tests and analyzes the practical application of the designed text clustering method, and finds that it contains 1219 texts, 75436 total type characters and 79,281 total class characters. The text material consists of seven sub-collections: Point of view, Around the world, In conversation, Latest news, Special report, Zoom, and Dossier. The experimental environment of this paper is Windows 10 64bit operating system, MatlabR2020b execution environment, and hardware environment is Intel (R) Core (TM) i7-8550U CPU@1.80 GHz, 16GB RAM laptop. In the experiment, eps is set to 1 and MinPts is set to 7.

The suggested method MO6, MO1 method [7], MO2 method [8], MO3 method [13], MO4 method [16] and MO5 method [18] were used to analyze the clustering accuracy of business English texts respectively, and the assessment index of clustering accuracy was adopted as the accuracy rate, and the experimental results of the clustering accuracy of different text categories are shown in Table 1. All six clustering methods reached the optimal value in Latest news text, indicating that the features of Latest news are easy to distinguish and facilitate accurate clustering. The accuracy of the offered MO6 method is higher than the other five methods on all seven business English text categories, and the average accuracy of MO6 is 91.2%, which is 17.6%, 10.3%, 8.3%, 6.4%, and 3.7% higher than MO1, MO2, MO3, MO4, and MO5, respectively. MO1 and MO2 are based on K-modes and K-means algorithms for text clustering, respectively, but these two algorithms have poor ability to deal with outliers, which affects the clustering effect. MO3 clusters text through the idea of semi-supervised learning, but does not consider the effect of common words on the clustering effect. MO4 optimizes the clustering effect by optimizing the distance of the density points, but it does not adequately extract the text features. MO5 is not optimized for TF-IDF text feature extraction, and is also optimized for DBSCAN, so the accuracy is not as good as MO6.

5.2. **Clustering effect analysis.** To comprehensively measure the clustering performance of the proposed MO6 method, this paper evaluates the clustering effect using seven commonly used metrics, namely, profile coefficient S [27], Davies-Bouldin index (DBI), adjusted mutual information (AMI), adjusted Rand coefficient (ARI) [28], homogeneity (H), and completeness (C) [29], which are shown in Figure 4.

Larger values of S, AMI, ARI, C and H indicate better clustering. In the evaluation index S, MO6 has the highest value of 0.5819 and MO1 has the lowest S value of 0.3759. In terms of the DBI evaluation index, the larger the DBI value, the worse the clustering effect, MO1 is the highest with a DBI of 0.6869, the DBI value of MO2 is the next highest with 0.6415, and the DBI of MO6 is the smallest with 0.2428, which indicates that the maximum similarity between the categories of MO6 is the lowest when clustering. In the

Table 1. Clustering accuracy of different business English text types

Categories	**Accuracy (%)**					
	MO1	**MO2**	**MO3**	**MO4**	**MO5**	**MO6**
Point of view	72.9	78.3	83.4	86.9	87.2	89.4
Around the world	70.2	83.6	80.9	83.2	86.1	88.3
In conversation	75.6	80.2	84.8	85.9	89.7	94.4
Latest news	75.8	84.8	85.6	87.8	90.5	95.1
Special report	74.1	81.5	80.6	82.5	85.4	91.6
Zoom	72.1	79.1	83.1	84.1	88.6	90.2
Dossier	74.7	78.9	81.7	83.7	85.3	89.1
Average accuracy	**73.6**	**80.9**	**82.9**	**84.8**	**87.5**	**91.2**

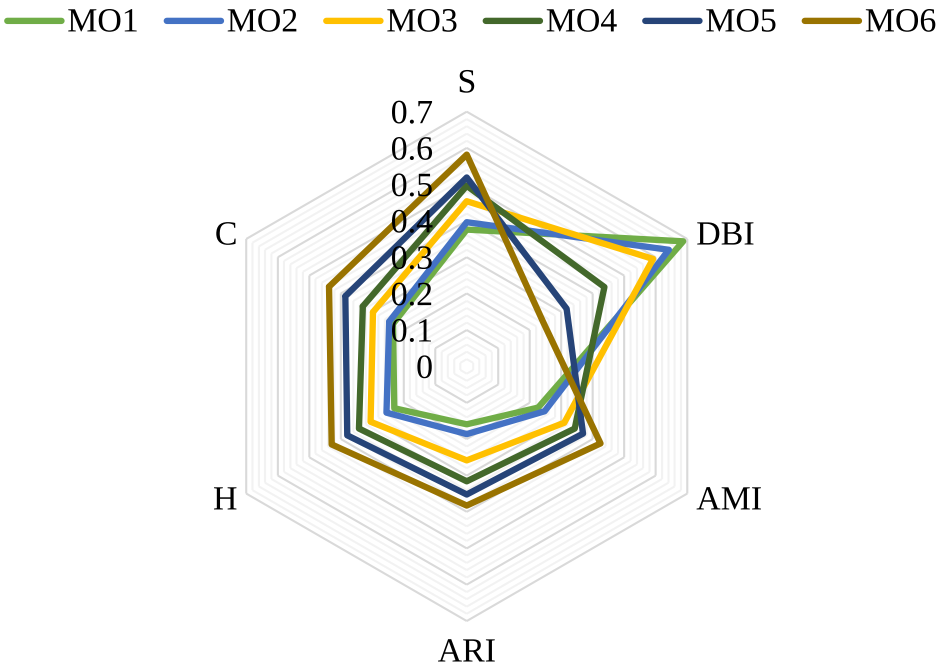

Figure 4. Comparison of clustering performance metrics

results of AMI and ARI, the value of MO6 is 0.4246 and 0.3826 respectively, which is the highest among the six clustering algorithms. The AMI and ARI of MO3 are in the middle with 0.3093 and 0.2584 respectively. In terms of H and C indexes, the H and C of MO6 are 0.4289 and 0.4371 respectively, which are still the highest among the three clustering algorithms. Therefore, the proposed MO6 method is more accurate in clustering, with better clustering results and better overall performance of the algorithm, fully reflecting the advantages of MO6.

Figure 5 shows the clustering effect of the six algorithms. although MO1 and MO2 can classify the data set into seven clusters, they cannot accurately distinguish the data points in each cluster, and the points in a cluster are mistakenly classified into different categories, which leads to poor clustering effect. MO3 can correctly identify the number of class clusters, but it is still insufficient in accurately clustering all data points, MO4

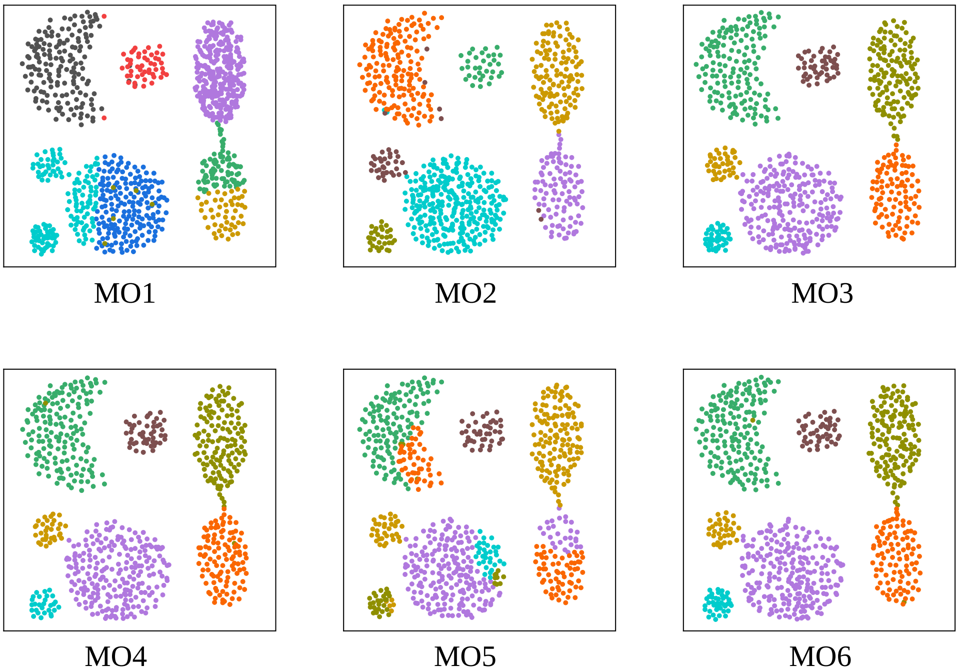

Figure 5. Comparison of clustering effects of six methods

and MO5 have fewer errors and cannot accurately identify the data points in the class clusters, so the clustering effect is not good, and MO6 can accurately identify all data points, so the clustering effect is the best.

6. **Conclusion.** In this paper, for the problem that the business English text clustering algorithm is sensitive to noisy data, which leads to poor clustering results, a business English text clustering method based on the improved DBSCAN algorithm is proposed. DBSCAN is optimized using weighted nearest neighbor assignment, which assigns the label of only one sample at a time, making full use of the information of the assigned clusters within the nearest neighbors, and does not affect the clustering results even if there is an error point. Skip-gram model is introduced to predict business English text context words to learn the distributional representation of text words, and the text features are extracted by TF-IDF algorithm optimized by inter-class and intra-class discretization. The similarity of texts is calculated based on the Hamming distance, and texts with similarity below a threshold are treated as isolated points. The EDBSCAN algorithm is utilized to cluster the texts, and similar texts are clustered into one class. The experimental outcome show that the clustering effect of the suggested method is better than other methods, and it can efficiently realize the clustering of business English texts.

Although the suggested method has achieved some research results, whether the $MinPts$ parameter in DBSCAN is set appropriately or not will also affect the final clustering effect, and how to find an optimal $MinPts$ parameter in practical use is an important research direction in the future.

Acknowledgment. This work is supported by the 2023 Guangdong Provincial Higher Vocational Colleges Curriculum Ideology and Politics Demonstration Course "Spoken English for International Trade" (No. KCSZ04249).

REFERENCES

[1] H. F. Martins, "Perspectives on business English as a lingua franca in business communication," *Teacher Education and Curriculum Studies*, vol. 2, no. 5, pp. 61-67, 2017.

[2] H. S. Sznajder, "A corpus-based evaluation of metaphors in a business English textbook," *English for Specific Purposes*, vol. 29, no. 1, pp. 30-42, 2010.

[3] P. Sunhare, R. R. Chowdhary, and M. K. Chattopadhyay, "Internet of things and data mining: An application oriented survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 3569-3590, 2022.

[4] E. S. Dalmaijer, C. L. Nord, and D. E. Astle, "Statistical power for cluster analysis," *BMC Bioinformatics*, vol. 23, no. 1, 205, 2022.

[5] S. Naeem, and A. Wumaier, "Study and implementing K-mean clustering algorithm on English text and techniques to find the optimal value of K," *International Journal of Computer Applications*, vol. 182, no. 31, pp. 7-14, 2018.

[6] M. A. Ahmed, H. Baharin, and P. N. Nohuddin, "Mini-batch k-means versus k-means to cluster english tafseer text: View of al-baqarah chapter," *Journal of Quranic Sciences and Research*, vol. 2, no. 2, pp. 48-53, 2021.

[7] G. Suryanarayana, L. Prakash K, P. S. Mahesh, and T. Bhaskar, "Novel dynamic k-modes clustering of categorical and non categorical dataset with optimized genetic algorithm based feature selection," *Multimedia Tools and Applications*, vol. 81, no. 17, pp. 24399-24418, 2022.

[8] P. Das, D. K. Das, and S. Dey, "A modified Bee Colony Optimization (MBCO) and its hybridization with k-means for an application to data clustering," *Applied Soft Computing*, vol. 70, pp. 590-603, 2018.

[9] J. V. Munoz, M. A. Gonçalves, Z. Dias, and R. d. S. Torres, "Hierarchical clustering-based graphs for large scale approximate nearest neighbor search," *Pattern Recognition*, vol. 96, 106970, 2019.

[10] S. Kongwudhikunakorn, and K. Waiyamai, "Combining distributed word representation and document distance for short text document clustering," *Journal of Information Processing Systems*, vol. 16, no. 2, pp. 277-300, 2020.

[11] X. Ran, Y. Xi, Y. Lu, X. Wang, and Z. Lu, "Comprehensive survey on hierarchical clustering algorithms and the recent developments," *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8219-8264, 2023.

[12] A. Lang, and E. Schubert, "BETULA: Fast clustering of large data with improved BIRCH CF-Trees," *Information Systems*, vol. 108, 101918, 2022.

[13] V. H. A. Soares, R. J. Campello, S. Nourashrafeddin, E. Milios, and M. C. Naldi, "Combining semantic and term frequency similarities for text clustering," *Knowledge and Information Systems*, vol. 61, pp. 1485-1516, 2019.

[14] R. G. Creţulescu, D. I. Morariu, M. Breazu, and D. Volovici, "DBSCAN algorithm for document clustering," *International Journal of Advanced Statistics and IT&C for Economics and Life Sciences*, vol. 9, no. 1, pp. 58-66, 2019.

[15] M. Li, X. Bi, L. Wang, and X. Han, "A method of two-stage clustering learning based on improved DBSCAN and density peak algorithm," *Computer Communications*, vol. 167, pp. 75-84, 2021.

[16] L. Yin, H. Hu, K. Li, G. Zheng, Y. Qu, and H. Chen, "Improvement of DBSCAN Algorithm Based on K-Dist Graph for Adaptive Determining Parameters," *Electronics*, vol. 12, no. 15, 3213, 2023.

[17] L. Abualigah, A. H. Gandomi, M. A. Elaziz, H. A. Hamad, M. Omari, M. Alshinwan, and A. M. Khasawneh, "Advances in meta-heuristic optimization algorithms in big data text clustering," *Electronics*, vol. 10, no. 2, 101, 2021.

[18] A. K. Sangaiah, A. E. Fakhry, M. Abdel-Basset, and I. El-henawy, "Arabic text clustering using improved clustering algorithms with dimensionality reduction," *Cluster Computing*, vol. 22, no. 2, pp. 4535-4549, 2019.

[19] M. A. Al-Betar, A. K. Abasi, G. Al-Naymat, K. Arshad, and S. N. Makhadmeh, "Bare-bones based salp swarm algorithm for text document clustering," *IEEE Access*, vol. 11, pp. 2169-3536, 2023.

[20] Q. Bsoul, R. Abdul Salam, J. Atwan, and M. Jawarneh, "Arabic text clustering methods and suggested solutions for theme-based quran clustering: analysis of literature," *Journal of Information Science Theory and Practice*, vol. 9, no. 4, pp. 15-34, 2021.

[21] S. Qaiser, and R. Ali, "Text mining: use of TF-IDF to examine the relevance of words to documents," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25-29, 2018.

[22] N. Hanafi, and H. Saadatfar, "A fast DBSCAN algorithm for big data based on efficient density calculation," *Expert Systems with Applications*, vol. 203, 117501, 2022.

[23] P. D. Turney, and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of Artificial Intelligence Research*, vol. 37, pp. 141-188, 2010.

[24] Y. Feng, C. Hu, H. Kamigaito, H. Takamura, and M. Okumura, “A simple and effective usage of word clusters for CBOW model,” *Journal of Natural Language Processing*, vol. 29, no. 3, pp. 785-806, 2022.

[25] S. Akbar, M. Hayat, M. Tahir, S. Khan, and F. K. Alarfaj, “cACP-DeepGram: classification of anticancer peptides via deep neural network and skip-gram-based word embedding model,” *Artificial Intelligence in Medicine*, vol. 131, 102349, 2022.

[26] D. D. Prasetya, A. P. Wibawa, and T. Hirashima, “The performance of text similarity algorithms,” *International Journal of Advances in Intelligent Informatics*, vol. 4, no. 1, pp. 63-69, 2018.

[27] K. Ntirogiannis, B. Gatos, and I. Pratikakis, “Performance evaluation methodology for historical document image binarization,” *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 595-609, 2012.

[28] J. A. Lossio-Ventura, S. Gonzales, J. Morzan, H. Alatrista-Salas, T. Hernandez-Boussard, and J. Bian, “Evaluation of clustering and topic modeling methods over health-related tweets and emails,” *Artificial Intelligence in Medicine*, vol. 117, 102096, 2021.

[29] M. Yuan, J. Zobel, and P. Lin, “Measurement of clustering effectiveness for document collections,” *Information Retrieval Journal*, vol. 25, no. 3, pp. 239-268, 2022.