Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

A Study of Word Vector Computation and Multilayer Network Representation in English Corpus

Chunmei Qiao[1,✉]

[1] *The Public Course Teaching Department, Henan Vocational University of Science and Technology, Zhoukou, Henan, 466000, China*

ABSTRACT

Although China's research on English is not as early as that of the western countries, researchers, combining the basic national conditions of China and the actual situation of the nationals' learning of English, have been making continuous efforts in the research on the construction and application of English corpus, and have already achieved satisfactory results. In this paper, we first analyze the related contents of English corpus, and construct English corpus corpus from phonological and semantic aspects by analyzing the correlation characteristics between English corpus and semantics, according to the basic principles of corpus selection. Combining two word vector similarity measures, Jaccard similarity and edit distance, finally constitutes the final similarity calculation algorithm for English sentences. The MECNC model is constructed by integrating the joint representation and co-representation learning methods, and using edge probability to abstract the connection between two nodes. Experimentally analyze the word vector similarity of English corpus with the results of English corpus recommendation based on multilayer network representation. The correlation scores of Jaccard similarity metric in WS-SIM, WS-REL, MEN, Mtruk-771, and Simverb-3500 are 0.8069, 0.6668, 0.7389, 0.7125, respectively, 0.2769, which achieves the best results, so Jaccard captures more of the correlation between words. Experiments on link prediction task were conducted on five corpora using 3, 5, 8, and 10-fold cross-validation methods, and on the corpus CKM [245,1550], MECNC model OM3 has a maximum AUC value close to 0.94 at a cross-validation number of 8, which shows that MECNC, which is used as a guiding information for intra-layer wandering, shows a better performance.

Keywords: associative features; MECNC model; English corpus; word vector computation; multilayer network representation

✉ Corresponding author.
E-mail address: qcmcassie821826@163.com (C. Qiao).
Received 03 February 2025; Revised 15 March 2025; Accepted 01 April 2025; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-008

1. Introduction

Corpus refers to a large number of well-sampled and processed electronic texts, which can be used to conduct theoretical or applied research on language with the help of computer tools, and are indispensable resources for corpus linguistics and empiricist language to conduct research [1-4]. And the English corpus brings together the real language data of the English language, provides more specialized and standardized research resources, and plays a pivotal role in various academic language studies in China [5-6]. Therefore, in-depth understanding and research on English corpus has become a top priority.

With the rise of artificial intelligence and big data research, natural language processing, as an interdisciplinary research integrating linguistics and computer science, has gained extensive attention from academia and industry [7-9]. The premise of natural language processing is text representation, i.e., how to convert human symbolic text into a form of representation that can be "understood" by computers [10-12]. Early natural language representations were mainly discrete. In recent years, with the continuous development of deep learning, neural network-based distributed word vector technology, based on algorithmic training of massive corpus, embeds symbolic sentences into a low-dimensional dense vector space, which shows strong potential and application effects in parsing syntax and analyzing semantics [13-16]. As one of the basic units for expressing semantics, words are the main objects of natural language processing. The basic concept of word vector is to numerically or vectorially characterize human symbolized words [17-19]. The current word characterization methods are mainly discrete and distributed [20].

In this paper, the Viterbi algorithm is used to input the correct English pronunciation method, sound recognition ability and English phoneme imitation ability into the corpus, meanwhile, the corpus is predicted in English according to the semantically related features of the words to complete the construction of the English corpus. The Jaccard similarity calculation algorithm based on word vectors is proposed to replace the co-occurrence of words with the degree of co-occurrence of word vectors. Combined with the edit distance algorithm, a comprehensive similarity calculation algorithm is composed to compute the word vectors in the corpus. Associate joint representation learning and collaborative representation learning to constitute a multilayer network representation in the English corpus, and on this basis, use edge probability to abstract the connection between two nodes, design the MECNC model, and apply the edge probability model to the multilayer network in a simple way. Simulation experiments are set up to examine the performance of word vector similarity computation of English words and English corpus recommendation under multilayer network representation learning, respectively.

2. Word Vector Computation and Multilayer Network Representation Learning in English Corpus

2.1. English corpus construction

In this paper, we first carry out the content analysis related to English corpus, and construct English corpus corpus according to the basic principles of corpus selection by analyzing the correlation characteristics between the semantics of English corpus corpus.

2.1.1. Construction of voice training. The correct pronunciation method, sound discrimination ability and English phoneme imitation ability will be inputted into the corpus, and the comprehensive training will be carried out through the Viterbi algorithm that is:

$$H = \frac{S}{GP_i}\left[\frac{1}{h_a}\left(\lambda_m - \lambda_n\right)\right] \tag{1}$$

Where, S is the correct pronunciation pitch of single phoneme for speech training, G is the pronunciation deviation threshold of single phoneme for speech training, P is the word pitch for speech training, h_a is the overall intonation for speech training, λ_m is the highest frequency of the output speech, and λ_n is the lowest frequency of the output speech.

The basic rules of English word stress and sentence stress, the teaching and training of expressive and ideographic functions, are specified by the formula:

$$P = \frac{\tau}{w_a}\left[\frac{1}{h_a}\left(\lambda_m - \lambda_n\right)\right] \cdot H \tag{2}$$

where w_a is the intensity value of linearly predicted accents, h_d is the intensity value of actual detected accents, and τ is the critical bandwidth of word accents.

Training expressions for English reading patterns:

$$A = \frac{\tau}{w_a}\left[\frac{1}{z_i}(P+H)\right] \tag{3}$$

where z_i is the number of strongly read syllables in the sentence.

Based on the above analysis, the construction of speech training in English corpus corpus is realized.

2.1.2. Semantic training construction. In order to build an English corpus, the corpus will be predicted in English according to the semantic association features of the words, which effectively enhances the activity, interactivity, and real-time of English learning, and Equation (4) represents the specific steps for the semantics of the corpus to be predicted [21]:

$$T_i = \frac{(P+H)}{\tau} \cdot l * A \tag{4}$$

In order to ensure that the difficulty of the corpus is in a coordinated relationship with the students' level, English materials that meet the students' interests are selected as the selected content in the English semantic training. The specific workflow of the English corpus corpus is shown in Figure 1.

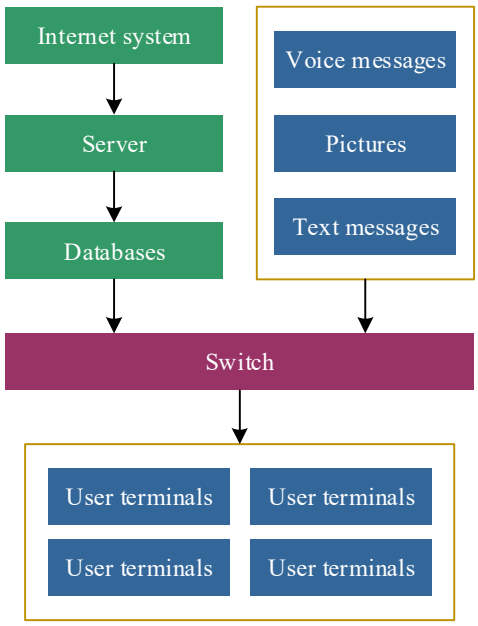


Fig. 1. Workflow of English corpus database

2.2. Word Vector Calculation in English Corpus

2.2.1. Jaccard similarity calculation algorithm based on word vectors. The traditional Jaccard similarity measure algorithm measures the similarity of two sentences or phrases based on the overlap of words [22]. The basic principle is that co-occurring words are the basic constituent units of sentence semantics, so the more co-occurring and overlapping the words are, the higher the semantic similarity of the two sentences will be. It is easy to see that a big problem of this method is that synonyms and near-synonyms are excluded from the semantic similarity calculation because they are not counted in the co-occurrence. The Jaccard similarity calculation algorithm based on word vectors proposed in this paper replaces the co-occurrence of words with the co-occurrence of word vectors, i.e., the degree of semantic co-occurrence. The formulaic expression is as follows:

$$\mathrm{Cos}\,Dis\left(w_1,w_2\right)=\frac{w_1\cdot w_2}{\left\|w_1\right\|\times\left\|w_2\right\|}=\frac{\sum_{i=1}^{n}w_{1i}\times w_{2i}}{\sqrt{\sum_{i=1}^{n}\left(w_{1i}\right)^2}\times\sqrt{\sum_{i=1}^{n}\left(w_{2i}\right)^2}} \tag{5}$$

$$SimScore_{Jaccard}\left(I,R\right)=\frac{\sum_{w\in I}\min\left(\max\left(\alpha\times\mathrm{Cos}\,Dis\left(w,R\right)\right),1\right)+\sum_{w\in R}\min\left(\max\left(\alpha\times\mathrm{Cos}\,Dis\left(w,I\right)\right),1\right)}{\left|I\right|+\left|R\right|} \tag{6}$$

In Equation (5), $\mathrm{Cos}\,Dis\left(w_1,w_2\right)$ is used to calculate the cosine similarity of word vectors w_1 and w_2 corresponding to two words, and n is the dimension of the word vector. In Eq. (6), I and R are the current sentence input by the user and the sentence retrieved by the system from the translation memory, respectively. $\max\alpha\times\mathrm{Cos}\,Dis\left(w,R\right)$ is the maximum value of the cosine similarity between the word vectors corresponding to all the words in sentence R and the word vector corresponding to w. Parameter α is the amplification factor used to adjust the cosine similarity between two word vectors, because the word vector models trained according to different corpora are different, and their calculation results will fluctuate, which can be set and adjusted according to the differences of the obtained word vector tables [23]. In order to prevent the amplification effect of the amplification coefficient from exceeding the actual representation range $\left[-1,1\right]$, a threshold is set for its maximum value.

2.2.2. Similarity calculation algorithm based on word vector and edit distance. Edit distance is often used in the similarity calculation of two sentences. However, the traditional method based on edit distance and the improved method based on edit distance and semantic lexicon both inevitably have a lot of shortcomings in semantic analysis, for which this paper applies word vectors to edit distance, and then obtains the similarity computation algorithm based on word vectors and edit distance used in this paper. Below is an example of how edit distance is applied to a sentence, calculating the edit distance of two sentences "I like computer science." and "I love computer technology." and their similarity.

The editorial distance based on word vectors is obtained above, and the formula for calculating sentence similarity is expressed as follows:

$$SimScore_{edit\_dis}\left(I,R\right)=1-\frac{edit\_dis\,\tan ce^{\max\left(\left|I\right|,\left|R\right|\right)}}{\max\left(\left|I\right|,\left|R\right|\right)} \tag{7}$$

2.2.3. Comprehensive similarity calculation. For the English language similarity calculation mainly consists of these two parts. Jaccard similarity based on word vectors mainly takes into account the degree of co-occurrence of related words, which has both superficial and lexical similarity. While the word vector-based edit distance not only considers the semantic, contextual relevance of the words themselves, but also the similarity of the sentence structure. Therefore, although the composition of the algorithm is relatively simple, it contains a variety of similarity calculation factors, combining the advantages of the two algorithms, and weighting and summing them to form the final similarity calculation algorithm for English sentences:

$$SimScore(I,R) = \rho_1 SimScore_{Jaccard}(I,R) + \rho_2 SimScore_{edii\_dis}(I,R) \tag{8}$$

Eq. $\rho_1 + \rho_2 = 1$, and satisfies $0 < \rho_2 \le \rho_1 < 1$.

2.3. Multilayer Network Representation in the English Corpus

Figure 2 shows a schematic diagram of joint representation learning and collaborative representation learning. Joint representation learning is to get a feature representation by directly fusing nodes from different layers together through an intermediate layer, i.e., all layers of the network share a single feature space, while co-representation learning is to exchange complementary information with other layers through an intermediate layer, and then each layer is individually embedded into its own vector space.

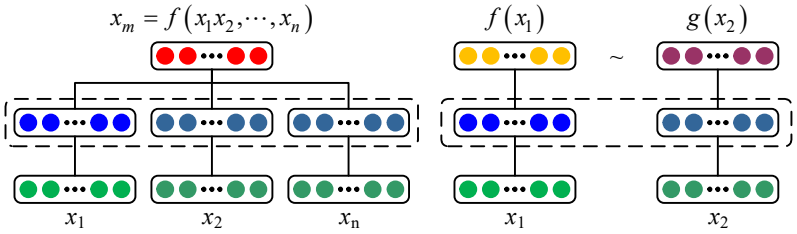


Fig. 2. Joint representation learning and collaborative representation learning schemes

2.3.1. Network Embedding Method Based on Edge Probability. In network representation learning, edge probabilities are commonly used to abstract the connection between two nodes.The MECNC method also employs a network embedding model based on edge probabilities.

Given a network, if there are edges between nodes v_i and v_j, then the edge probability between them is defined as:

$$p(v_i, v_j) = \frac{1}{1 + \exp(-v_i^T, v_j)} \tag{9}$$

where v_i and v_j are the vector representations of nodes v_i and v_j, respectively. If the two nodes have edges, the value of p should be close to 1. Otherwise, the value of p should be close to 0. Thus, learning the vector representations of the nodes becomes generating the original network with maximum likelihood, and the likelihood is computed as follows:

$$\prod_{(v_i, v_j) \in E} p(v_i, v_j) \prod_{(v_i, v_j) \notin E} \left(1 - p(v_i, v_j)\right) \tag{10}$$

where E denotes the set of edges. For ease of computation, the loss function is usually log-likelihood with a regularization term of the node vector matrix V to avoid overfitting. The loss function is defined as follows:

$$Loss(\varepsilon;\lambda) = -\sum_{(v_i,v_j)\in E} \log p\left(v_i v_j\right) - \sum_{(\omega_i,v_j)\notin E} \log\left(1-p\left(v_i,v_j\right)\right) + \lambda\left\|V\right\|_F^2 \tag{11}$$

where row i of the node vector matrix $V \in R^{N\times d}$ is the vector representation of the i rd node v_i, λ are the regularization coefficients.

In reality, the network size is usually very large, and each iteration when optimizing Eq. (11) needs to compute N^2 edge probabilities, which is very computationally intensive, and the number of $\left(v_i,v_j\right)\notin E$ is much larger than that of $\left(v_i,v_j\right)\in E$. This imbalance will lead to the features obtained from the model learning to be more inclined to negative samples. To solve this problem, negative sampling is usually used, i.e., sampling $\left(v_i,v_j\right)\notin E$ negative samples, the number of negative samples is defined as $\left|E_{neg}\right| = k\left|E_{pos}\right|$, where E_{pos} is the set of positive sample edges, and k is the sampling rate. Therefore, equation (11) can be rewritten in the following form:

$$Loss\left(E_{pos};E_{neg};\lambda\right) = -\sum_{(v_i,v_j)\in E_{pos}} \log p\left(v_i v_j\right) - \sum_{(v_i,v_j)\in E_{neg}} \log\left(1-p\left(v_i,v_j\right)\right) + \lambda\left\|V\right\|_F^2 \tag{12}$$

By minimizing the loss function (Eq. 12), all nodes of the network are embedded in a d -dimensional vector space. Finally, computing $p\left(v_i,v_j\right)$ then predicts whether edges exist for v_i and v_j.

2.3.2. MECNC model design:

1) Multilayer Network Edge Probabilistic Modeling. The purpose of the multilayer network edge probability model is to capture the proximity of nodes encoded in each layer, so for each node v_i, this paper uses a layer-specific representation $\left\{v_i^l\right\}_{l=1}^L$ to preserve the structural information of the node in each layer. For an edge $\left(v_i^l,v_j^l\right)$ in layer l, the edge probability is defined as follows with reference to equation (9):

$$p\left(v_i^l,v_j^l\right) = \frac{1}{1+\exp\left(-v_i^{ll},v_j^l\right)} \tag{13}$$

From this with reference to equation (12), the loss function of the edge probability model of the multilayer network can be defined as follows:

$$Loss\left(E_{pos},E_{neg};\lambda\right) = \sum_{t=1}^{L}\left(-\sum_{\left(v_i^l,v_j^l\right)\in E_{pos}^l} \log p\left(v_i^l,v_j^l\right) - \sum_{\left(v_i^l,v_j^l\right)\in E_{neg}^l} \log\left(1-p\left(v_i^l,v_j^l\right)\right)\right) + \lambda\left\|V\right\|_F^2 \tag{14}$$

2) Multilayer Collaboration Based on Node Contexts. Simply applying the edge probability model to the multilayer network will not play the role of inter-layer synergy to promote each other, because at this time the embedding space of each layer is still independent of each other. A more accurate representation of nodes can only be obtained by fusing the structures of different layers into the same semantic space.

An intuitive approach to multilayer spatial fusion is to introduce a layer of intermediate bridges that connect the embedding spaces of each layer, thus achieving multilayer synergy. Specifically,

MECNC achieves this by adding a multilayer spatial synergy term to the loss function. MECNC splits the representation of each node into two roles: one is responsible for learning the structural features of the node (both the features of the current layer and fusing the features of the other layers), and each layer is independent of each other and is referred to as a layer-specific node representation. The other plays the role of a bridge for connecting all layer spaces, shared by all layers, called node context representation. The mathematical form of the synergy term is basically the same as Eq. (14):

$$Coordinated = \alpha \sum_{i=1}^{L} \left(- \sum_{\left(v_i^l, v_j^l\right) \in E_{pos}^l} \log p\left(v_i^l v_j^c\right) - \sum_{\left(v_i^l, v_j^l\right) \in E_{neg}^l} \log\left(1 - p\left(v_i^l, v_j^c\right)\right) \right) + \beta \left\| C \right\|_F^2 \tag{15}$$

where v_j^c denotes the context of node j, which is shared by all layers, $C \in R^{N \times d}$ is the representation matrix of node contexts, α denotes the coefficients of the synergetic terms, and β is the coefficients of the regularization of the node context vectors. Since the node context is directly connected to each specific layer node representation, thus the node representations of different layers share the semantic space, and the node context is connected to the specific layer nodes through the set of positive and negative sample edges, so that v_i^l is also able to learn the structural features of the other layers to achieve the purpose of inter-layer synergy.

3) Inter-layer node feature synergy enhancement. Through node contexts, the semantic space of multilayer networks can be effectively connected and the structural features of other layers can be synergistically learned. However, this synergy effect is limited because the layer-specific representations of nodes are similar only if the local structures (first-order proximity) of the same nodes in different layers are similar. In contrast, in some multilayer networks, the semantics of each layer of the network is very close to each other, even though the local structures of the nodes are not the same. For example, in a social network, if the interpersonal relationships in a department are layered in terms of casual, work, and micro-relationships, the semantics of each relationship does not differ much from the overall view, even though the direct connection of each person in each relationship may differ, and if the semantic spaces of all the layers can be fused together, it is expected that the synergy between the multilayered networks can be further improved.

To address this issue, MECNC introduces an augmentation term in the loss function, specifically, if the semantic differences between layers are not large, then the distances of the node-specific layer representations in the vector space should also be close, i.e., the variance of the node-specific layer representations v_i^l should be as small as possible:

$$\text{var} = \frac{1}{L} \sum_{l=1}^{L} \left\| v_i^l - \overline{v}_i \right\|_F^2 \tag{16}$$

where $\overline{v}_i$ denotes the mean value of node i represented in all layers. Expanding Eq. (16) to a multilayer network, the mathematical expression of the enhancement term is shown below:

$$\left(The\ augmented\ term\right) = \gamma D[V] \tag{17}$$

$$D[V] = \frac{1}{L} \sum_{l=1}^{L} \left\| V[l] - E[V] \right\|_F^2 \tag{18}$$

$$E[V] = \frac{1}{L} \sum_{l=1}^{L} V[l] \tag{19}$$

Where, γ is the enhancement term coefficient, $V[l]$ denotes the $N \times d$ node vector matrix of layer l, L, N, d denotes the number of layers, nodes and embedding vector dimensions, respectively, $E[V]$ is the mean value of the node vector matrices of all layers, and $D[V]$ is the mean squared deviation of the node vector matrices of all layers.

4) Loss function design. It is important to note that in directed graphs $p\left(v_i, v_j\right) \neq p\left(v_j, v_i\right)$ and therefore Eq. (14) only applies to undirected graphs. In order to make MECNC applicable to both directed and undirected graphs, in this paper, an undirected edge is split into two directed edges, which are represented by two vectors, namely, the out-degree vector and the in-degree vector. As a result, $p\left(v_i^l, v_j^l\right)$ is redefined as:

$$p\left(v_i^l, v_j^l\right) = \frac{1}{1+\exp\left(-v_{Hi}^{lT}, v_{Tj}^l\right)} \tag{20}$$

In summary, after simultaneously considering the multilayer network edge probabilities, node contextual synergy, enhancement terms, and node out-degree and in-degree representations, the loss function of the model is designed as follows:

$$\begin{aligned} & Loss\left(E_{pos}, E_{neg}; \lambda, \alpha, \beta, \gamma\right) \\ & = -\sum_{\left(v_i^l v_j^l\right) \in E_{pos}} \log p\left(v_i^l, v_j^l\right) - \sum_{\left(v_i^l v_j^l\right) \in E_{neg}} \log\left(1 - p\left(v_i^l, v_j^l\right)\right) \\ & -\alpha\left(\sum_{\left(v_i^l, v_j^l\right) \in E_{pos}} \log p\left(v_i^l, v_j^c\right) - \sum_{\left(v_i^l v_j^l\right) \in E_{neg}} \log\left(1 - p\left(v_i^l, v_j^c\right)\right)\right) \\ & +\gamma\left(D\left[V_H\right] + D\left[V_T\right]\right) \\ & +\frac{\lambda}{N}\left(\left\|V_H\right\|_F^2 + \left\|V_T\right\|_F^2\right) + \frac{\beta}{N}\left(\left\|C_H\right\|_F^2 + \left\|C_T\right\|_F^2\right) \end{aligned} \tag{21}$$

where $E_{pos} = E^1 \cup \ldots \cup E^L$ denotes the set of all edges in the multilayer network and E_{neg} is the set of negatively sampled edges, obtained by uniformly randomly sampling from all unconnected pairs of nodes in the multilayer network with a sampling rate of k.

The optimal node representations V_H and V_T are learned by minimizing the loss function Loss, and finally link prediction is performed using Eq. (20). Due to the use of node in/out representations, Eq. (20) can be directly applied to directed graphs. For undirected graphs, the set of positive sample edges is learned as $E_{pos} = E^1 \cup \ldots \cup E^L \cup E' \cup E'^L$, where $E' = \left\{\left(v_i^l, v_j^l\right) \mid \left(v_j^l, v_i^l\right) \in E^l\right\}$, and then the average of $p\left(v_i^l, v_j^l\right)$ and $p\left(v_j^t, v_i^t\right)$ is used to predict whether edge $\left(v_i^t, v_j^t\right)$ exists or not.

3. Experimental setup and analysis

3.1. Computational analysis of word vectors in the English corpus

3.1.1. English Word Visualization. In this paper, we select Wikipedia Dump of English in 2017 as the training corpus of English polysemous word vectors, take the first 50,000 words according to the word frequency statistics from high to low, and calculate the co-occurrence of each word with each other, and the window is set to 10 to construct the PPMI matrix to carry out the NMF factor decomposition, and the dimensions of the decomposition are, for $d = 100$, $H[:, i]$ for the word w_i of the 100-dimensional low-rank expression as the SFCM input, set the clustering categories as C 100~1 000 for qualitative analysis, and use the DBI index to select the optimal number of clusters. In order to reduce the number of clusters for word polysemous words, the SFCM algorithm parameters are set, $m = 1.01, k = 2, \zeta = 0.05$, that is, each word will not have more than 2 meanings, and it is

found through experiments that the larger the value is set, the greater the risk that the semantics of certain polysemous words will be overly refined, resulting in an overly large dictionary space V'.

DBI is a metric for evaluating clustering algorithms, which serves to assess the effectiveness in clustering algorithms based on the dispersion of clustered data points. The DBI is computed as follows, first a value of dispersion S_i is defined to indicate the degree of dispersion of the ind cluster metric data point:

$$S_i = \left\{ \frac{1}{T_i} \sum_{j=1}^{T_i} \left| X_j - A_i \right|^q \right\}^{1/q} \tag{22}$$

Where A_i denotes the center of cluster i, X_j denotes the jth data point in cluster i, T_i denotes the number of data points in cluster i, q taken as 1 denotes the evaluation of the distance from each point to the center, and q taken as 2 denotes the standard deviation of the distance from each point to the center.

Secondly, a distance value M_{ij} is defined to denote the distance between cluster i and cluster j.

$$M_{ij} = \left\{ \sum_{k=1}^{N} \left| a_{ki} - a_{kj} \right|^p \right\}^{1/p} \tag{23}$$

Where a_{ki} denotes the krd eigenvalue of the centroid of cluster i,where p denotes the different metric distances,in this lab $p = 2$ denotes the Euclidean distance.

Thirdly,the similarity value $R_{ij} = \frac{S_i + S_j}{M_{ij}}$ of cluster i and cluster j is defined,and then the largest similarity value in R_{ij} is selected,i.e.:

$$R_i = \max(R_{ij}) \tag{24}$$

Finally, the DBI index of the clustering result is obtained, where C is the number of clusters clustered:

$$R = \frac{1}{C} \sum_{i=1}^{C} R_i \tag{25}$$

The number of suitably clustered clusters is quantitatively analyzed by setting different C values. Fig. 3 shows the DBI scores for different number of clusters, the two folds in the figure indicate the DBI values for 100~1000 classes at q=1, q=2 for the categories of C respectively. It can be seen from the figure that the minimum DBI for 500 clusters, q=1, q=2 values are 0.51042 and 0.57519 respectively. The results show that C=500 is the best choice both from the evaluation of the center distance of the points and from the evaluation of the standard deviation.

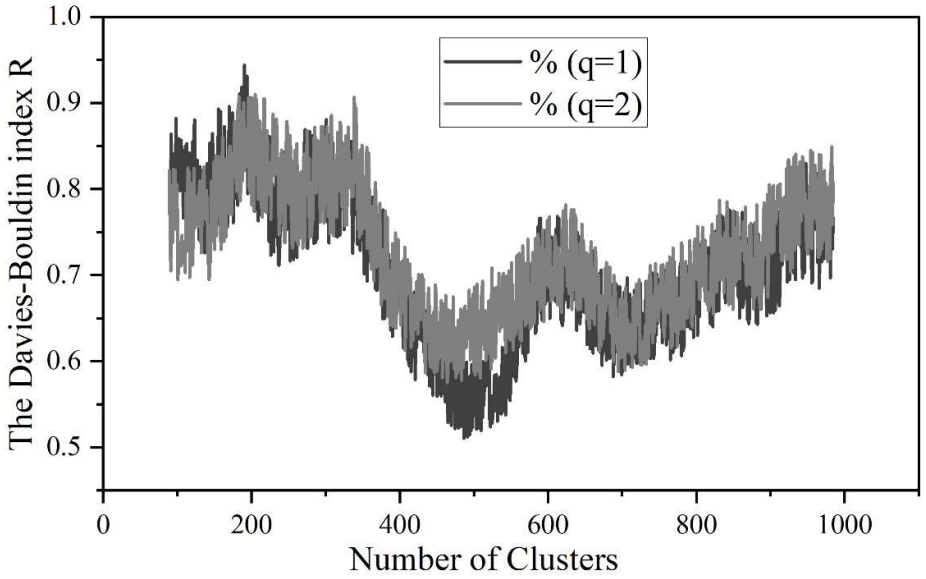
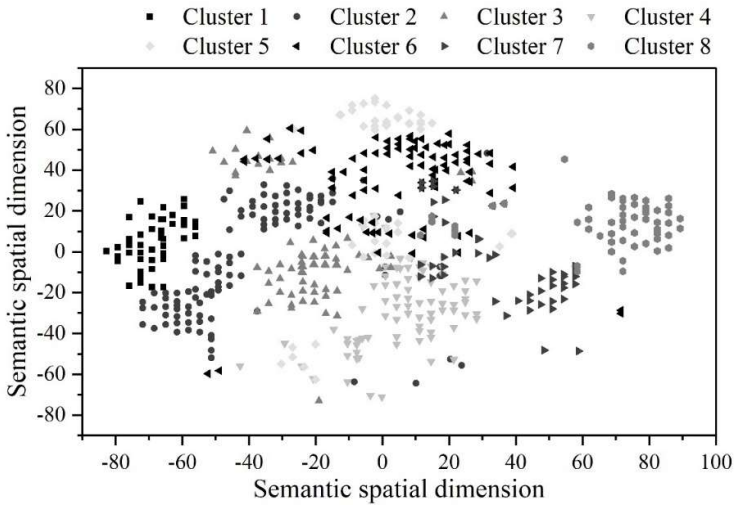


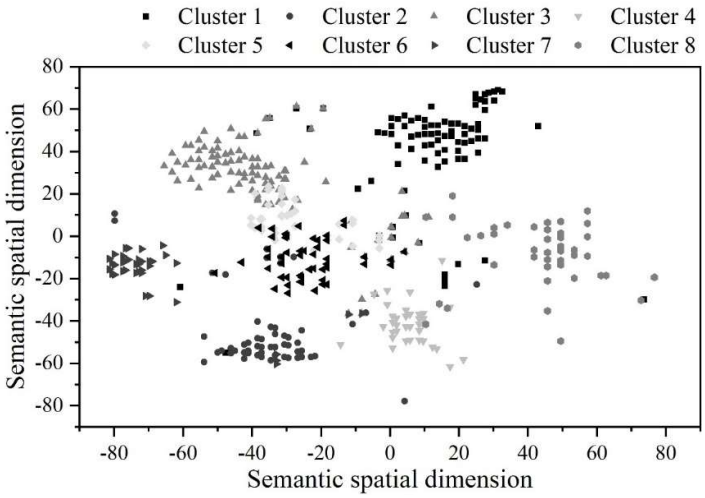
Fig. 3. DBI scores of different clusters

In order to visualize the clustering more intuitively, three sets of values ($C = 200,500,1000$) were also selected, which represent the cases of different cluster sizes. The low-rank representations of 100-dimensional words are visualized into the 2D plane by t-SNE stream mapping, and Fig. 4 shows the scatters of different numbers of clusters for sparse soft clustering, with C=200 in Fig. (a), C=500 in Fig. (b), and C=1000 in Fig. (c).(a)~(c) are the clustering hashmaps for the cases, respectively, and the background of where the scatters are located in randomly selected 8 clusters are shown to the streaming space is the semantic space.

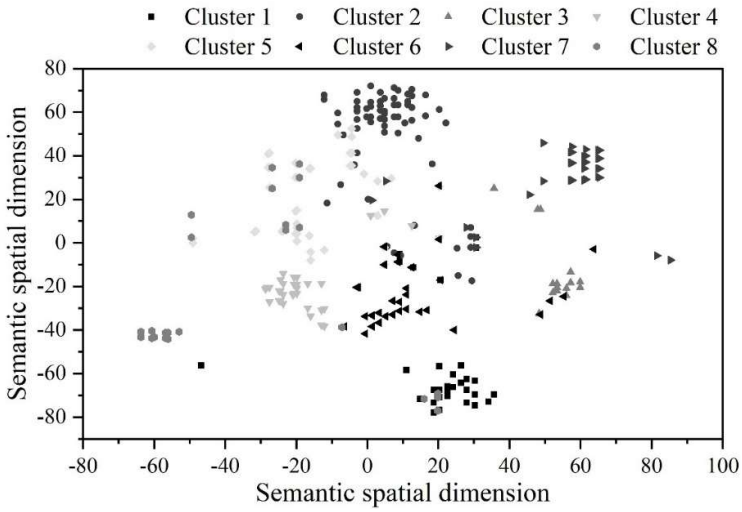
It is found that the number of co-occurrences between words directly affects the spatial representation of low-rank expressions, and the low-rank expression vectors of polysemous words after NMF decomposition have more non-zero elements than the low-rank expressions of monosemous words, reflecting the fact that monosemous words are in a lower-dimensional subspace than polysemous words. This phenomenon is shown in the figure as follows: most of the monosemantic words are clustered at the periphery of the whole semantic space and are more compact. Polysemantic words are mostly located in the center of the clusters and appear sparser. From the size of the clusters, Fig. 4(a), when $C = 200$, the affiliation of some word clusters under the Jaccard similarity measure becomes the average value because the affiliation is not effectively distinguished during the iteration process, some of the cluster centers are collapsed towards the semantic center (geometric center), and the identification of polysemous words becomes exceptionally difficult, and Fig. 4(c), when $C = 1000$, the excessive clusters lead to the number of words in each cluster is too small. The clusters that should not be divided are divided, and the lack of vocabulary dictionaries is difficult to meet the number of embedding training, the experiment found that $C = 500$ is more appropriate, in Fig. 4(b), the number of vocabulary dictionaries in each cluster is between 36 and 98, at the same time, the clustering centers are more clear, and there is no collapse towards the semantic center.



(a)C=200



(b)C=500



(c)C=1000

Fig. 4. Sparse soft clustering different number clustering

3.1.2. Word similarity experiments. Relying on the word similarity and relevance task as an intrinsic assessment of word vectors, the correlation coefficient between human judgments and the cosine similarity between word vectors is used as a judgment. This calculation is fast and easy to understand, and it is important to know explicitly which class of features Jaccard is capturing, so a range of datasets were collected for the similarity and relevance rubrics.

The datasets independently tested for word similarity include WS-SIM, Simlex-999, and those tested for relevance include WS-REL, MEN, MTurk-771, as well as the test set WS-353, Simverb-3500, which does not distinguish between the two properties.

Table 1 shows a comparison of the Spearman rank correlation coefficients between the cosine scores of the different models and the human ratings. The results were chosen as Jaccard comparisons since independent words cannot be contextually identified with their specific labeled clusters.

It is observed from the table that Jaccard's correlation scores for WS-SIM, WS-REL, MEN, Mtruk-771, and Simverb-3500 outperform the other models by reaching the best (0.8069, 0.6668, 0.7389, 0.7125, 0.2769), which contains all three correlation test sets. Thus Jaccard captures more of the correlation between words.

Table 1. The correlation coefficient between cosine similarity

Model	WS-SIM	Simlex-999	WS-REL	MEN	Mturk-771	WS-353	Simverb-3500
Word2vec	0.7554	0.3795	0.5412	0.6848	0.5953	0.6425	0.2648
Glove	0.7966	0.3705	0.6315	0.7066	0.6515	0.7198	0.2298
Lexvec	0.7345	0.3615	0.6612	0.7215	0.6632	0.6815	0.2584
FastText	0.7955	0.4069	0.6632	0.7236	0.6395	0.7396	0.2698
Jaccard	0.8069	0.3485	0.6668	0.7389	0.7125	0.7369	0.2769

3.1.3. Text classification. A set of text classification experiments are designed in which the continuous feature representation of word vectors as inputs in supervised learning is crucial and directly affects the classification results in a text classification task. The experiments are conducted by changing the input of static pre-trained word vectors and observing the classification accuracy to judge the merits of the model. Firstly, text categorization tasks of different sizes and attributes are selected, including news categorization task (20NG), question and answer categorization (TREC), sentiment categorization task (IMDB, SST), and Table 2 shows the description of the English dataset used in the sentence categorization task.

Table 2. Description of the English datasets used in sentence classification tasks

Data set	Number of classes	Average sentence length	Training sample	Test sample
20NG	25	436	15987	3915
IMDB	3	245	25365	25365
TREC	5	12	5423	503
SST1	4	20	11866	2216
SST2	2	20	9645	1832

Table 3 shows the accuracy of the model under the English test set, and Jaccard outperforms all other models in the news classification task (20NG), question and answer score (TREC), and sentiment classification tasks (IMDB, SST1). $Jaccard_{multi}$ achieved the highest accuracy in the IMDB, TREC, and SST1 tasks, with accuracies of 0.8736, 0.9408, and 0.4498, respectively. In the 20NG task, $Jaccard_{single}$ improved its accuracy by at least 5.79% over the other models. The results show that $Jaccard_{multi}$ outperforms $Jaccard_{single}$ in the short text categorization task. In these tasks, short

texts provide less information compared to long texts, and word vectors of polysemous words in short texts can take full advantage of the fine-grained semantics of Jaccard's model, whereas $Jaccard_{single}$ word vectors are more sensitive to semantic patterns in long texts, as evidenced by the 20NG task. Jaccard outperforms word2vectors in all tasks. Outperforms word2vec, GloVe and LexVec, except for the sentiment task SST2, where it only slightly underperforms the Fasttext model.

Another advantage of the Jaccard similarity computation is that it can be applied in the form of $Jaccard_{multi}$ or $Jaccard_{single}$ depending on the text length of the task to achieve the desired performance.

Table 3. Precision of the model in the English test set

Model	20NG	IMDB	TREC	SST1	SST2
Word2vec	0.7165	0.8615	0.9345	0.4469	0.7958
Glove	0.7398	0.8716	0.9152	0.3645	0.8165
Lexvec	0.7264	0.8625	0.8536	0.3815	0.8266
FastText	0.7766	0.8715	0.9315	0.3769	0.8516
$Jaccard_{single}$	0.8345	0.8621	0.9285	0.4465	0.8241
$Jaccard_{multi}$	0.7896	0.8736	0.9408	0.4498	0.8436

3.2. English corpus recommendation for multilayer network representation learning

3.2.1. Experimental corpus. Table 4 shows the statistics of the multilayer web corpus, listing basic information about the dataset, with more detailed information as follows:

Table 4. Multi-layer network corpus statistics

Data set	Number of floors	Number of nodes	Edge number
Vickers	3	30	740
Kapferer	4	40	1015
CKM	3	245	1550
C.ELEGANS	3	280	5860
FF-TW-YT	3	6405	74860

3.2.2. Methods of comparison. In order to fully demonstrate the effectiveness of this model on multilayer networks for complex network analysis tasks, classical single-layer network representation learning models (DeepWalk, Node2vec, and LINE) and mainstream multilayer network representation learning models (MNE, mGCN, IFMNE, and NIFMNE) are chosen for comparison.DeepWalk: sampling in a randomized wandering manner generates sequences of nodes in the network and as sentences, inputs them into a skip-gram model, and trains to generate node vectors.

Node2vec: control the random walk by introducing a pair of parameters p and q so that its neighborhood can be obtained by depth-first traversal or breadth-first traversal.

LINE: LINE adds the direct link fitting term to the DeepWalk loss function and adds the second-hop buddy to the first-hop buddy to merge higher-order relationships.

MNE: MNE can efficiently store and learn multiple types of relationship information and then embed them uniformly.

mGCN: mGCN can extract interaction information from within and between dimensions.

IFMNE: A multilayer network representation learning method that fuses node structure information and spatial structure information, and introduces the parameter r to guide random wandering across layers.

NIFMNE: A basic version of IFMNE that considers only node structure information to guide random wandering.

3.2.3. Experimental setup. A common link prediction method is based on node similarity, where a similarity score s_{ij} is computed for each pair of nodes $< v_i, v_j >$ in the network, and then links that do not appear in the network are ranked based on this score, with links with higher similarity scores considered more likely to exist. For the node vectors that have been embedded by network representation learning, the cosine similarity between the vectors is usually used to calculate the node similarity. The cosine similarity between vector X and vector Y is calculated as shown below:

$$\cos(X,Y) = \frac{X^T Y}{|X| \cdot |Y|} \tag{26}$$

In this paper, we use the AUC score as an evaluation metric for the results of link prediction experiments, which is a common evaluation metric for link prediction and recommender systems, and can reflect the accuracy of link prediction among nodes. Each time from the test set E^p and the non-existent edge set $U - E$ each randomly select an edge, according to the cosine similarity of the embedding vector to calculate their similarity scores for comparison, assuming that in n independent comparisons, there are n' times in the test set of edge scores are higher, there are n'' times the two scores are the same, then the AUC scores for this experiment:

$$AUC = \frac{n' + 0.5n''}{n} \tag{27}$$

The extent to which the AUC score exceeds 0.5 can reflect how much better the algorithm used performs than pure random selection, so for all the model experimental results the higher the AUC score indicates better performance and higher quality embedding vectors that reflect the node's characteristics in the original network.

3.2.4. Experimental results and analysis. In this paper, experiments on the link prediction task were conducted on five corpora using 3-, 5-, 8-, and 10-fold cross-validation methods, respectively. Each time, the corpus is divided into a corresponding number of copies, and then one of them is sequentially selected as the test set and the rest as the training set for the experiment and the results are averaged. The experimental results are shown in Table 5-Table 8.

1) 3-fold cross-validation. The AUC scores of each method at the number of 3-fold cross-validation are shown in Table 5, and MECNC performs second only to IFMNE on the FF-TW-YT corpus, and achieves optimal results on the rest of the corpora. Among them, the version of OM1 using first-order local structural entropy has the best results, with performance improvement ranging from 2.4% to 5.9% on different corpora compared to the rest of the optimal performance.

Table 5. 3 Double cross verification

Corpus	Vickers	Kapferer	CKM	C.ELEGANS	FF-W-YT
DeepWalk	0.7465	0.7765	0.7615	0.8345	0.9055
Node2vec	0.7566	0.7185	0.7452	0.8469	0.9163
LINE	0.6648	0.5615	0.6245	0.6728	0.8166
MNE	0.7585	0.7068	0.8846	0.8615	0.9165
mGCN	0.7516	0.7958	0.8315	0.8846	0.7063
IFMNE	0.7816	0.7548	0.8755	0.8695	0.9245
NIFMNE	0.8069	0.7615	0.8813	0.8615	0.8863
OM1	0.8348	0.8196	0.9315	0.8845	0.9193

OM2	0.8059	0.7798	0.8815	0.8654	0.9175
OM3	0.8079	0.7925	0.9302	0.8966	0.9185

2) 5-fold cross-validation. The performance of each method at 5-fold cross-validation counts is shown in Table 6, and MECNC achieved good results on different corpora. Among them, the version of OM1 using first-order local structural entropy works best on two smaller corpora, Vickers and Kapferer, with AUC values of 0.8345 and 0.8265, respectively, and the version of OM3 using third-order local structural entropy achieves optimal performance on the remaining three larger corpora, with AUC values of 0.9358, 0.9056 and 0.9246.

Table 6. 5 double cross verification

Corpus	Vickers	Kapferer	CKM	C.ELEGANS	FF-W-YT
DeepWalk	0.7615	0.8054	0.7955	0.8615	0.9348
Node2vec	0.7765	0.7136	0.7816	0.8726	0.9248
LINE	0.7166	0.5816	0.6645	0.7165	0.8236
MNE	0.7632	0.7298	0.8896	0.8615	0.9248
mGCN	0.7516	0.7958	0.8513	0.8854	0.6945
IFMNE	0.8063	0.7763	0.8816	0.8766	0.9435
NIFMNE	0.7955	0.7615	0.8796	0.8769	0.9054
OM1	0.8345	0.8265	0.9334	0.8955	0.9026
OM2	0.8169	0.7826	0.8925	0.8766	0.9164
OM3	0.8183	0.7956	0.9358	0.9056	0.9246

3) 8-fold cross-validation. The performance of each method at the number of 8-fold cross-validation is shown in Table 7. For the vast majority of multilayer networks, the performance of the multilayer network representation learning methods that incorporate information from different layers outperforms the models designed for single-layer networks (DeepWalk, LINE et al.). Second, the most significant difference of the model proposed in this paper is the introduction of local node local influence information and inter-layer variability information as a guide index for random wandering, as well as cross-layer wandering sampling, fusing richer structural information from different layers, which achieved optimal or sub-optimal performances in the above corpus, with an improvement of between 0.03% and 27% compared to other methods. This shows that integrating local and overall structural information of multilayer networks is effective.

Table 7. 8 double cross verification

Corpus	Vickers	Kapferer	CKM	C.ELEGANS	FF-W-YT
DeepWalk	0.7758	0.8066	0.8164	0.8846	0.9245
Node2vec	0.7869	0.7345	0.7982	0.8815	0.9266
LINE	0.7293	0.5795	0.6648	0.7062	0.8815
MNE	0.7687	0.7296	0.9045	0.8752	0.9245
mGCN	0.7536	0.8045	0.8645	0.8822	0.7169
IFMNE	0.8166	0.7796	0.8926	0.8841	0.9345
NIFMNE	0.8065	0.7596	0.8912	0.8836	0.8936
OM1	0.8369	0.8246	0.9345	0.9055	0.9158
OM2	0.8169	0.7736	0.8866	0.8831	0.9056
OM3	0.8226	0.7953	0.9348	0.9083	0.9348

4) 10-fold cross-validation. Table 8 shows the 10-fold cross-validation, the model proposed in this paper improves from 0.8% to 24.31% compared to other methods. This shows that the combined consideration of local and overall structural information of multilayer networks is effective.

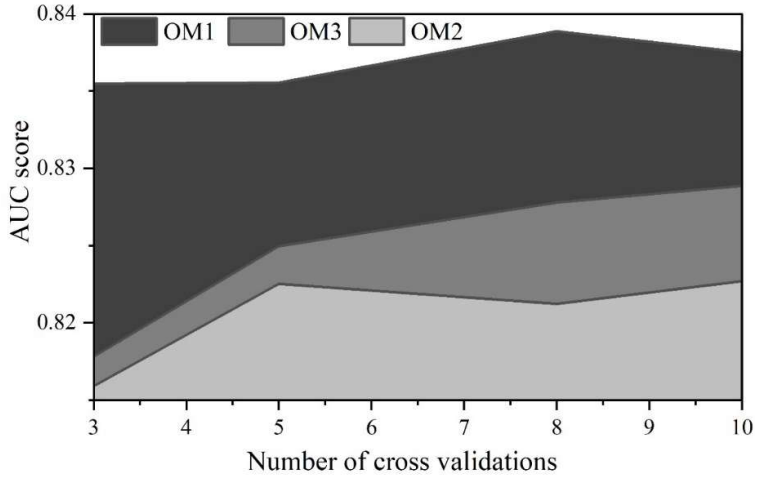
Table 8. 10 double cross verification

Corpus	Vickers	Kapferer	CKM	C.ELEGANS	FF-W-YT
DeepWalk	0.7765	0.8054	0.8135	0.8812	0.9315
Node2vec	0.7715	0.7436	0.8063	0.8869	0.9375
LINE	0.7296	0.5796	0.7063	0.7025	0.8836
MNE	0.7798	0.7395	0.9015	0.8736	0.9348
mGCN	0.7523	0.8036	0.8836	0.8826	0.7045
IFMNE	0.8135	0.7798	0.8896	0.8816	0.9476
NIFMNE	0.8063	0.7766	0.8874	0.8805	0.9126
OM1	0.8375	0.8215	0.9385	0.9042	0.9264
OM2	0.8166	0.7816	0.8966	0.8835	0.9158
OM3	0.8269	0.8065	0.9305	0.9136	0.9388

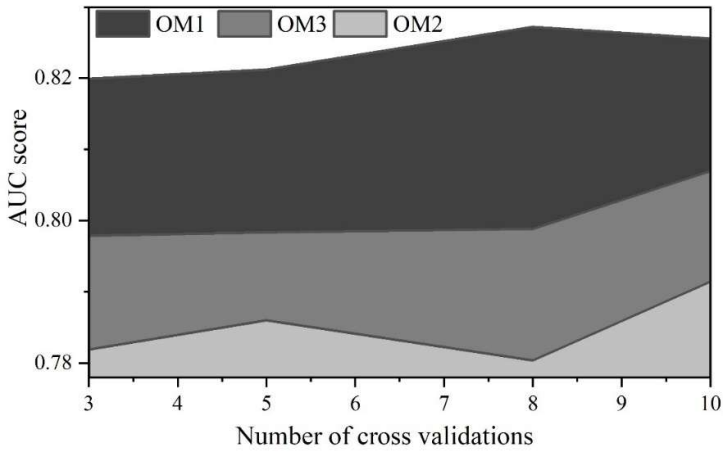
5) Link task prediction. Figure 5 shows a comparison of the link prediction task scores of different versions of MECNC on five corpora, Figure (a) for Vickers [30,740], Figure (b) for Kapferer [40,1015], Figure (c) for CKM [245,1550], Figure (d) for clegans [280,5860], and Figure (e) for FF-TW- YT [6405,74860].

As the size of the corpus grows, MECNC with higher order local structural entropy (OM2, OM3) as in-layer wandering guidance information shows better performance. On corpus CKM [245,1550], OM3 gradually outperforms OM1, with the maximum value of AUC approaching 0.94 at a cross-validation count of 8. Subsequently, OM3 has the highest value of AUC in telegans [280,5860] and FF-TW-YT [6405,74860], which are 0.91001 at a cross-validation count of 10 and 0.93821.

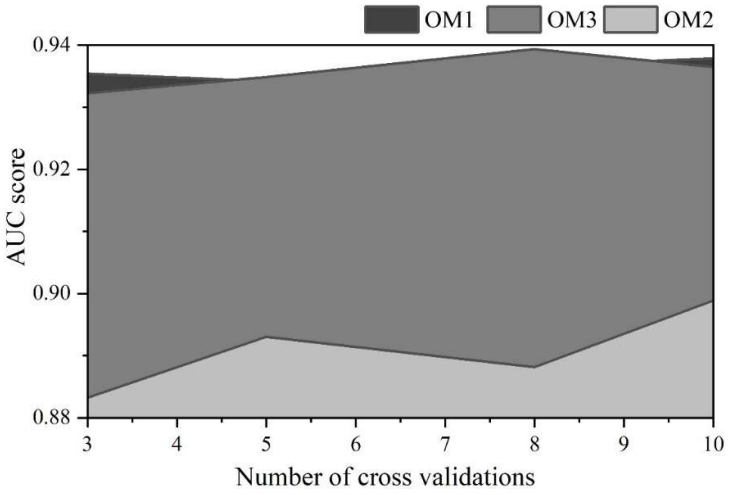
This is due to the fact that the higher-order local structural entropy better reflects the node's influence in a wider range, which is used as a bootstrap information to better reflect its local characteristics in a larger network. However, for a specific multilayer network, how many orders of local structural entropy can be used as the guidance information for random wandering within the layer to obtain the optimal performance needs to be verified by several experiments.



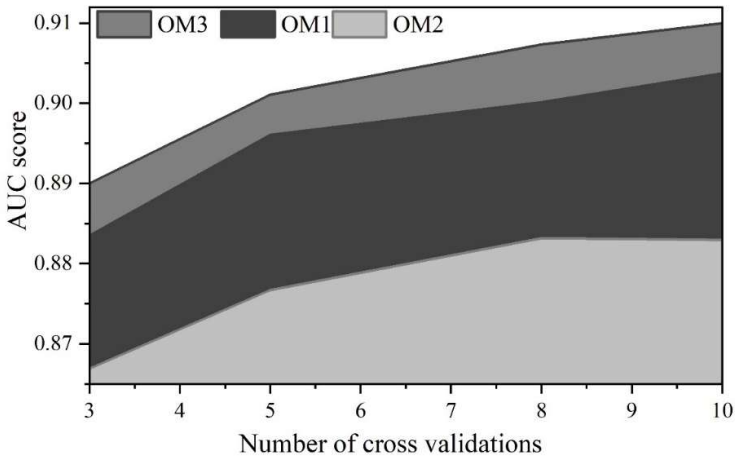
(a)Vickers[30,740]



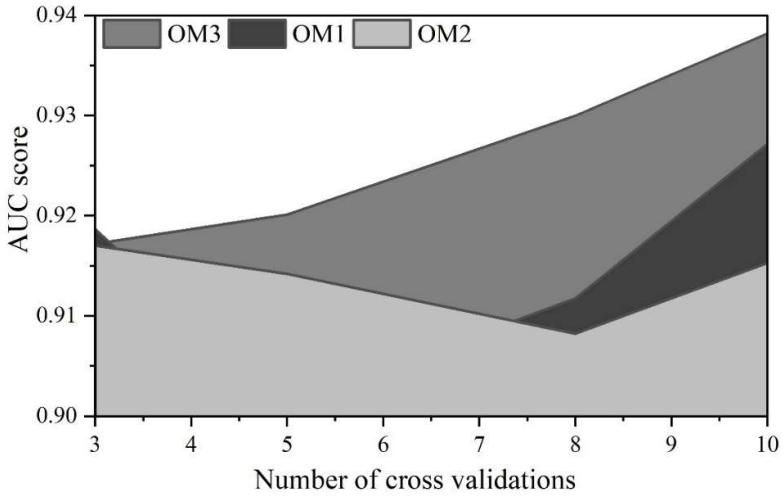
(b)Kapferer[40,1015]



(c)CKM[245,1550]



(d)celegans[280,5860]



(e)FF-TW-YT[6405,74860]

Fig. 5. Link prediction task score contrast

4. Conclusion

In this paper, we train an English speech corpus by inputting the correctly pronounced English phoneme imitation ability into the corpus through the Viterbi algorithm. The corpus will be predicted in English according to the semantic association features of words. Combining the advantages of Jaccard similarity measure algorithm and edit distance, the similarity calculation of English sentences is finally obtained. Combine two multilayer network representations, joint and collaborative, to complete the design of MECNC model. Simulation experiments are set up to calculate word vectors in English corpus with the results of corpus recommendation under the learning of multilayer network representations. The relevance scores of Jaccard's similarity calculation algorithm in WS-SIM, WS-REL, MEN, Mtruk-771, and Simverb-3500 outperform the other models and reach the best, respectively, with the similarity values of 0.8069, 0.6668, 0.7389, 0.7125, and 0.2769, and Jaccard better captures the correlation between words. The performance of the algorithm is evaluated by the AUC metric, and the model is cross-validated with 3, 5, 8, and 10 re-validations, and in the 10 re-validations, the model proposed in this paper improves from 0.8% to 24.31% compared to other methods. This shows the effectiveness of integrating the local and overall structural information of multilayer networks.

Acknowledgement

This work was supported by Innovation and Practice of an Educational Model for Vocational Undergraduate Public Basic Courses Emphasizing Both Moral and Technical Training, Industry-Education Integration, and Integrated Assessment, Henan Provincial Vocational Education Teaching Reform Project (2024.05858).

References

[1] Nurmukhamedov, U., & Sharakhimov, S. (2023). Corpus-based vocabulary analysis of English podcasts. RELC Journal, 54(1), 7-21.

[2] Anderson, W., & Corbett, J. (2017). Exploring English with online corpora. Bloomsbury Publishing.

[3] De Haan, P. (2024). Postmodifying clauses in the English noun phrase: A corpus-based study (Vol. 3). Brill.

[4] Esfandiari, R., & Barbary, F. (2017). A contrastive corpus-driven study of lexical bundles between English writers and Persian writers in psychology research articles. Journal of English for Academic Purposes, 29, 21-42.

[5] Sazzed, S., & Jayarathna, S. (2019, July). A sentiment classification in bengali and machine translated english corpus. In 2019 IEEE 20th international conference on information reuse and integration for data science (IRI) (pp. 107-114). IEEE.

[6] Hedberg, N., Sosa, J. M., & Görgülü, E. (2017). The meaning of intonation in yes-no questions in American English: A corpus study. Corpus Linguistics and Linguistic Theory, 13(2), 321-368.

[7] Ni, J., Li, J., & McAuley, J. (2019, November). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP) (pp. 188-197).

[8] Schmidt, A., & Wiegand, M. (2017, April). A survey on hate speech detection using natural language processing. In Proceedings of the fifth international workshop on natural language processing for social media (pp. 1-10).

[9] Kreimeyer, K., Foster, M., Pandey, A., Arya, N., Halford, G., Jones, S. F., ... & Botsis, T. (2017). Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review. Journal of biomedical informatics, 73, 14-29.

[10] Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019, June). Transfer learning in natural language processing. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials (pp. 15-18).

[11] Galassi, A., Lippi, M., & Torroni, P. (2020). Attention in natural language processing. IEEE transactions on neural networks and learning systems, 32(10), 4291-4308.

[12] Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. Science China technological sciences, 63(10), 1872-1897.

[13] Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., & Matsuo, A. (2018). quanteda: An R package for the quantitative analysis of textual data. Journal of Open Source Software, 3(30), 774-774.

[14] Coppersmith, G., Leary, R., Crutchley, P., & Fine, A. (2018). Natural language processing of social media as screening for suicide risk. Biomedical informatics insights, 10, 1178222618792860.

[15] Belinkov, Y., & Glass, J. (2019). Analysis methods in neural language processing: A survey. Transactions of the Association for Computational Linguistics, 7, 49-72.

[16] Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). Adversarial attacks on deep-learning models in natural language processing: A survey. ACM Transactions on Intelligent Systems and Technology (TIST), 11(3), 1-41.

[17] Camburu, O. M., Rocktäschel, T., Lukasiewicz, T., & Blunsom, P. (2018). e-snli: Natural language inference with natural language explanations. Advances in Neural Information Processing Systems, 31.

[18] Barnbrook, G. (2019). Language and computers: A practical introduction to the computer analysis of language. Edinburgh University Press.

[19] Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., ... & Hon, H. W. (2019). Unified language model pre-training for natural language understanding and generation. Advances in neural information processing systems, 32.

[20] Liu, X., Shin, H., & Burns, A. C. (2021). Examining the impact of luxury brand's social media marketing on customer engagement: Using big data analytics and natural language processing. Journal of Business

research, 125, 815-826.

[21] Wenxinxin Qin. (2024). A Corpus-Based Analysis of Pauses in Chinese-English Consecutive Interpreting of Chinese English Majors. English Language Teaching and Linguistics Studies(6),

[22] Arnold Hien,Noureddine Aribi,Samir Loudni,Yahia Lebbah,Abdelkader Ouali & Albrecht Zimmermann. (2024). Mining diverse sets of patterns with constraint programming using the pairwise Jaccard similarity relaxation. Constraints(prepublish),1-32.

[23] Junjie Lin,Xiaogang Wang,Mingrui Chang,Zhiwei Yin & Liuhong Zhang. (2024). A cosine similarity-based maximal clique point cloud registration algorithm.Computing(1),36-36.