Formulaic language identification model based on GCN fusing associated information

Fanqi Meng¹,², Yujie Zheng¹, Songbin Bao³, Jingdong Wang¹ and Shuaisong Yang¹

¹School of Computer Science, Northeast Electric Power University, Jilin City, Jilin, China
²School of Information Engineering, Guangdong Atv Academy For Performing Arts, Dongguan, Guangdong, China
³School of Foreign Language, Northeast Electric Power University, Jilin City, Jilin, China

ABSTRACT

Formulaic language is a general term for ready-made structures in a language. It usually has fixed grammatical structure, stable language expression meaning and specific use context. The use of formulaic language can coordinate sentence generation in the process of writing and communication, and can significantly improve the idiomaticity and logic of machine translation, intelligent question answering and so on. New formulaic language is generated almost every day, and how to accurately identify them is a topic worthy of research. To this end, this article proposes a formulaic language identification model based on GCN fusing associated information. The innovation is that each sentence is constructed into a graph in which the nodes are part-of-speech features and semantic features of the words in the sentence and the edges between nodes are constructed according to mutual information and dependency syntactic relation. On this basis, the graph convolutional neural network is adopted to extract the associated information between words to mine deeper grammatical features. Therefore, it can improve the accuracy of formulaic language identification. The experimental results show that the model in this article is superior to the classical formulaic language identification model in terms of accuracy, recall and F1-score. It lays a foundation for the follow-up research of formulaic language identification tasks.

INTRODUCTION

A formulaic language is a language segment composed of several words to be learned and used as an independent item, and it is a multi-word combination with high frequency (Strobl et al., 2019). It includes a wide variety of categories and functions, such as filler words (e.g., sort of), simple functional phrases (e.g., excuse me), word collocations (e.g., have a break), idioms or proverbs (e.g., spill the beans), and beginning of longer sentences (e.g., there is a growing body of evidence that). Formulaic language identification, also known as multi-word expression identification, refers to finding formulaic language that meets a certain definition from a large amount of corpus, and is also an essential task.
in natural language processing. The use of formulaic language can not only coordinate sentence generation in the process of writing and communication, improve the accuracy and fluency of language expression, but also have important theoretical and practical significance for computer-assisted language teaching and machine translation (Metin, 2018).

In recent years, the study of formulaic language has been in the ascendant. Scholars have made many research results on formulaic language with the help of corpus technology and computer applications AntConc and AntGram. However, there are some problems such as insufficient recognition standards, low recognition accuracy and efficiency, so recognizing formulaic language efficiently and accurately has become more critical. At present, formulaic language identification methods mainly include the statistics-based method, rule-based method and machine learning method (Hu, Yao & Gao, 2019). The identification method based on statistics and rules depends on the pre-set standards and has poor portability. It cannot effectively recognize many types of formulaic languages in the face of complex text. With the rise of machine learning in the field of natural language processing, some scholars try to use Random Forest (RF), Support Vector Machine (SVM) and other classifiers to identify formulaic language through classification technology. However, this method has high requirements for feature selection (Meng, Cheng & Wang, 2021). It needs to select a feature set that can effectively reflect the characteristics of formulaic language, resulting in its poor generalization ability.

In order to solve the above problems, this article proposes a formulaic language identification model based on GCN fusing associated information. The main technical contributions are summarized as follows:

- This model uses the embedding layer in Torch to generate word embedding vectors as part-of-speech features and uses feature vectors trained by GloVe word vector technology as semantic features. Part-of-speech features and semantic features are represented by Bi-LSTM respectively, and then the two features are fused. The fused feature vector is the basic feature of the formulaic language recognition model, which can fully represent the characteristics of high frequency of co-occurrence and fixed structure of formulaic language.

- This model inputs the associated information (including mutual information and dependency) between words and basic features into the Graph Convolutional Network (GCN) for feature representation. The obtained features can effectively represent the grammatical structure of formulaic language so that the relationship between words can be fully utilized. Finally, the high-order neighbor information between words can be captured.

- The high-order neighbor information is input to the CRF layer for decoding so that the problem of formulaic language recognition is regarded as a sequence labeling problem. This method can obtain the label category of each character, achieve the purpose of recognizing formulaic language, and provide a new idea for solving formulaic language recognition problems.
• We designed three groups of comparative experiments, and the experimental results proved the effectiveness of the proposed model. This model can recognize formulaic language in the text, which lays a foundation for expanding formulaic language corpus in machine translation.

The rest of this article is organized as follows. Section 2 reviews related work. Section 3 describes a formulaic language identification model based on GCN fusing associated information, including basic feature extraction module, feature extraction module fusing associated information and label representation module. Section 4 verifies the effectiveness of the model through experimental analysis. Section 5 summarizes the work of this article and tells the shortcomings and future research directions.

RELATED WORKS
At present, the research on formulaic language is either descriptive research based on the corpus or confirmatory research for language acquisition. There are very little researches related to the recognition of formulaic language. In fact, formulaic language recognition is the basis of applied research on formulaic language (Yu, 2019). However, manually recognizing formulaic language consumes many human resources and material resources, and the recognition efficiency is low. In order to improve efficiency, many scholars have carried out a series of research work on automatic recognition methods.

Formulaic language identification mainly adopts statistics-based method, rule-based method, machine learning method and deep learning method. Statistics-based recognition methods mainly use corpus technology to extract target word bundles from the corpus according to pre-set frequency and distribution standards. In the recognition standard, commonly used corpus parameters include frequency, Mutual Information, $t$-value, dispersion rate, criticality, etc. (Li, 2012). The corpus parameter used by Pecina is the phrase frequency, which judges whether several words can form a formulaic language by the frequency of co-occurrence of words (Pecina, 2009). However, if the frequency is used as the only criterion to identify formulaic language, it will ignore some formulaic language composed of low-frequency words with clear textual function, but extract too many phrases composed of high-frequency words without clear textual function. On this basis, the corpus parameters adopted by Li include not only the frequency but also the mutual information value (Lei & Liu, 2018).

Different from the statistical recognition method, the rule-based recognition method sets some part-of-speech and syntactic rules in advance to extract formulaic language (Kirilin, Krauss & Versley, 2016). Iwatsuki, Boudin & Aizawa (2020) believe that a sentence is composed of a formulaic part and an unformulated part, so they proposed to use named entities and dependent structures to remove the unformulated part of the sentence, and the rest is the formulaic part. At the same time, string matching and regular expressions can also be used to identify formulaic language (Durrant & Mathews-Aydnl, 2011). Through multiple linear regression analysis, Wahl (2019) found that the number of rules is positively correlated with the types of recognized formulaic language. The more rules are made, the
more types of formulaic language are recognized. However, our rules cannot contain all formulaic language, so the rule-based recognition method is very limited.

With the development of machine learning, Hidden Markov Model (HMM), Support Vector Machine (SVM), Conditional Random Field (CRF), Maximum Entropy (ME) and so on are used by researchers to recognize formulaic language. Abbas et al. (2020) chose to use RF and SVM classifiers to recognize formulaic language through classification. Hyland, Chen, Simpson-Vlach, etc. (Hyland, 2008; Chen & Baker, 2010; Simpson-Vlach & Ellis, 2010) proposed to extract N-grams from the corpus, but they can only extract formulaic language with consecutive words. Therefore, Moreau, Alsulaimani & Maldonado (2018) proposed two systems for recognizing formulaic language. CRF-DepTree-categs implements a dependency tree-based method, which aims to use the syntax and semantic relationship between tags; CRF-Seq-nocategs implements a sequence labeling method, which only needs lemma and morphological syntax tags. The experiment found that the latter recognizes the formulaic language with continuous words better, and the former recognizes the formulaic language with discontinuous words better.

With the development of deep learning, Gharbieh, Bhavsar & Cook (2017) tried to use CNN to recognize formulaic language. And they used word2vec training feature vectors. Ashok, Elmasri & Natarajan (2019) compared the role of word2vec and GloVe in recognizing formulaic language on the basis of Gharbieh, and found that GloVe is slightly better in recognizing formulaic language. Taslimipoor & Rohanian (2019) designed a model combining CNN and LSTM. At the same time, in order to deal with the problem of data scarcity in English data sets, firstly, the model is trained on the language with large data, and the learned knowledge is used to predict the tags of formulaic language in English. Subsequently, Taslimipoor, Bahaadini & Kochmar (2020) also designed a system that relies on the pre-trained language model of BERT to achieve semi-supervision of the system through multi-task learning.

The recognition method based on statistics can effectively recognize formulaic language with high frequency of co-occurrence. However, its main problem is that it does not consider the meaning of multi-word combinations and only takes frequency or correlation as the basis of discrimination, so the accuracy of recognition is low. The rule-based recognition method can accurately identify the formulaic language whose form is consistent with the rules defined in the model, which has the problem that the rules are not comprehensive enough. With the development of machine learning, some scholars try to use classifiers such as RF and SVM to recognize formulaic language through classification technology. However, it is necessary to extract features that can represent samples (Meng et al., 2022). The more appropriate the feature selection, the higher the accuracy of recognition. Therefore, feature selection is an urgent problem to be solved (Klyueva, Doucet & Straka, 2017).

To sum up, aiming at the problem of formulaic language recognition, this article proposes a formulaic language identification model based on GCN fusing associated information. This model represents the feature vector through word embedding technology, integrates the associated information that can represent the characteristics of formulaic language, and uses GCN to obtain deeper semantic features. The features obtained in this way are
closer to the formulaic language than the features extracted manually in machine learning. Finally, considering the dependency between tags, CRF is used to decode tags to achieve the purpose of recognizing formulaic language. This model not only takes into account the frequency of word co-occurrence, but also considers the dependencies of words in sentences, and achieves good results in formulaic language recognition.

**MODEL**

The model in this article is mainly divided into three parts: basic feature extraction module, feature extraction module fusing associated information and label representation module. Its overall structure is shown in Fig. 1. First, the text’s semantic features and part-of-speech features are extracted and fused through late fusion. The fusion results are used as the basic features of the model. Then the mutual information between words is calculated, the dependency syntactic parsing of sentences is analyzed, and the generated adjacency matrix and basic features are input into GCN for feature representation. Finally, the feature vector is input to the CRF layer for decoding to obtain the label category of each character, so as to obtain the formulaic language.

**Basic feature extraction module**

In the process of natural language processing, the computer cannot directly use text data. The text data needs to be expressed as a feature vector, and then the feature vector is used as the model’s input. This article uses the word embedding vector generated by the embedding layer in Torch as the part-of-speech feature, the feature vector trained by GloVe word vector technology as the semantic feature, and the late fusion of the part-of-speech and semantic features as the basic features of the model.

**Feature selection**

The most significant difference between formulaic language and general multi-word expression is that the structure of formulaic language is mostly fixed, which is often in the form of Verb-noun or subject–predicate–object. Therefore, part-of-speech features are regarded as one of the features to identify formulaic language. First, we use the Stanford part-of-speech tagger to analyze the text. The examples of part-of-speech analysis results are shown in Table 1. From the table, it can be found that multi-word units with fixed sentence patterns are more likely to be formulaic language. Then we assign a unique code to each result after part-of-speech tagging, so as to convert the text data into vectors. Finally, we input them into the embedding layer for training to generate word embedding vectors as part-of-speech features.

In addition, for text processing, characters, words, phrases more reflect the lexical information of the text rather than its semantic information, so they can not accurately express the content of the text. For formulaic language, it is a multi-word unit with high frequency and has complete structure, meaning and function, so semantic feature is also an essential feature of formulaic language. This article uses the feature vector trained by GloVe to represent the semantic features of formulaic language.
Figure 1 The overall structure of the model.

Table 1 The examples of Part-of-speech analysis result.

| Formulaic language                              | Part-of-speech tagging | Sentence structure                  |
|------------------------------------------------|------------------------|-------------------------------------|
| X is fundamental to                            | NN VBZ JJ TO           | Subject-Link verb-Predicative Structure |
| X plays a vital role in the metabolism of       | NN VBZ DT JJ NN IN DT NN IN | Subject Verb Object Structure       |
| Several attempts have been made to             | JJ NNS VBP VBN VBN TO | Subject Verb Object Structure       |
| In this innovative study, Smith showed that Y  | IN DT JJ NN FW FW NN VBD IN NN | Clauses guided by “that”            |
| There were some negative comments about Y      | EX VBD DT JJ NNS IN NN | There be...                          |
GloVe’s full name is Global Vectors for Word Representation. It is a word representation tool based on global word frequency statistics. It can express a word as a vector composed of real numbers. These vectors capture some semantic characteristics between words (Pennington, Socher & Manning, 2014). Its implementation is divided into the following three steps:

(1) A co-occurrence matrix $X$ is constructed according to the corpus. Each element $X_{ij}$ in the matrix represents the number of times that word $i$ and context word $j$ appear together in a context window of a specific size. Generally speaking, the minimum unit of this number is 1, but GloVe doesn’t think so: according to the distance $d$ between two words in the context window, it proposes a decreasing weighting function: $\text{decay} = \frac{1}{d}$ to calculate the weight. That is, the farther the distance, the smaller the weight of the two words in the total count.

(2) Construct the approximate relationship between Word Vector and Co-occurrence Matrix, as shown in Eq. (1).

$$w_i^T \tilde{w}_j + b_i + \tilde{b}_j = \log(X_{ij}).$$  \hspace{1cm} (1)

Among them, the $w_i^T$ and $\tilde{w}_j$ in the above formula are the word vectors we finally need to solve; and $b_i$ and $\tilde{b}_j$ are the bias terms of the two-word vectors.

(3) Construct the loss function, as shown in Eq. (2).

$$J = \sum_{i,j=1}^{V} f(X_{ij}) (w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij}))^2.$$  \hspace{1cm} (2)

Among them, $f(X_{ij})$ is the weight function. Its calculation formula is shown in Eq. (3).

$$f(x) = \begin{cases} (x/x_{\text{max}})^\alpha & \text{if } x < x_{\text{max}} \\ 1 & \text{otherwise} \end{cases}$$  \hspace{1cm} (3)

Among them, $x$ represents the number of co-occurrences, and $x_{\text{max}}$ represents the maximum number of co-occurrences.

**Feature representation based on Bi-LSTM**

Because LSTM is good at capturing the long-distance and long-term dependence of sentence context information, it can better avoid the problem of gradient disappearance and gradient explosion and has higher computational efficiency. Nevertheless, LSTM cannot capture the two-way information of sentences. For the task of formulaic language identification, if the forward information and backward information of sentences are added, the model can learn more semantic information when processing text (Hochreiter & Schmidhuber, 1997). Therefore, we use Bi-LSTM to learn the hidden layer representation of input sequences, hoping to obtain the features of sentences that can contain deeper semantic and syntactic information.

For sentence $s_i = [x_1, x_2, \ldots, x_t, \ldots, x_n]$, input it into the Bi-LSTM network, we can get the current hidden vector $h_t$ according to the calculation of the previous hidden vector $h_{t-1}$ and the current input vector $x_t$, and its operation is defined as follows:
\[ i_t = \sigma (W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \] (4)

\[ f_t = \sigma (W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \] (5)

\[ c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \] (6)

\[ o_t = \sigma (W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \] (7)

\[ h_t = o_t \tanh(c_t). \] (8)

In the formula: \( i_t, f_t, c_t, o_t, h_t \) are the state of input gate, forget gate, cell state, output gate and hidden layer when the \( t \)-th text is input; \( W \) is the parameter of the model; \( b \) is the bias vector; \( \sigma \) is the Sigmoid function; \( \tanh \) is the hyperbolic tangent function.

Bi-LSTM model is composed of forward LSTM and reverse LSTM models. The LSTM network of each layer outputs a hidden state information respectively, and the parameters of the model are updated by back propagation. The structure of Bi-LSTM model is shown in Fig. 2.

In Fig. 2, \( x_t \) represents the input of the network at time \( t \), the LSTM in the box is the standard LSTM model, \( \bar{y}_t \) is the output of the forward LSTM at time \( t \), and \( \hat{y}_{t-1} \) is the output of the reverse LSTM at time \( t \). \( \oplus \) indicates splicing operation. The output representation of
Feature fusion

Feature fusion includes early fusion and late fusion. Early fusion is to fuse multi-layer features first and then train the model on the fused features (only after complete fusion, unified training). In this article, early fusion is used as a comparative experiment. First, the part-of-speech features and semantic features are fused, and then the fused features are input into Bi-LSTM. The generated results are used as the basic feature vector. The structure diagram is shown in Fig. 3.

Compared with early fusion, late fusion first trains the model with a single feature and then fuses the training results of multiple models. The advantage of late fusion method is that it can flexibly select the model’s results and improve the fault tolerance of the system. The amount of calculation of fusion information is reduced and the real-time performance of the system is improved (Khaleghi, Khamis & Karray, 2013). In this article, the late fusion method is adopted. First, the part-of-speech features and semantic features are input into Bi-LSTM respectively, and then the results of the two models are spliced to form the basic feature vector. The structure diagram is shown in Fig. 4.

Feature extraction module fusing associated information

The basic feature extraction module uses the word embedding model to train in large-scale text. It can obtain word vectors rich in text semantic features and part-of-speech features, but the syntactic structure information in the text is ignored. As the basis of language understanding, the syntactic structure can effectively represent the grammatical structure of the text and reveal the relationship between the components of the text. For formulaic language, it is a multi-word unit with high frequency of co-occurrence, and several words with high correlations may form formulaic language. Therefore, it is very important to select the features that can represent the relationship between words for identifying formulaic language. Based on this, this article uses the mutual information between words and the
dependency syntactic relation of sentences as the associated information of formulaic language.

**Associated information based on mutual information**

Mutual information measures the correlation between two random variables, that is, the amount of information about another random variable contained in one random variable (Zhang & Wang, 2016). The mutual information of two discrete random variables X and Y is defined as:

\[
I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)},
\]

(9)

where \(p(x,y)\) is the joint probability distribution function of X and Y, \(p(x)\) and \(p(y)\) are the edge probability distribution functions of X and Y, respectively. If we want to measure the correlation degree of any two words \(x\) and \(y\) in a data set, we can calculate it as follows:

\[
I(x; y) = p(x,y) \log \frac{p(x,y)}{p(x)p(y)},
\]

(10)

where \(p(x)\) and \(p(y)\) are the probability of independent occurrence of \(x\) and \(y\) in the data set. It can be obtained by directly counting the word frequency and dividing it by the total number of words; \(p(x,y)\) is the probability that \(x\) and \(y\) appear in the data set at the same time. Directly count the number of times they appear at the same time, and then divide it by the number of all unordered pairs. Using mutual information to calculate the relationship between binary words, the higher the mutual information, the higher the correlation between \(x\) and \(y\), and the greater the possibility of forming formulaic language.

**Associated information based on dependency syntactic parsing**

Dependency syntax reveals the dependency relationship and collocation relationship between words in a sentence. One dependency relationship connects two words, one is the core word and the other is the modifier. This relationship is related to the semantic
relationship of the sentence. The dependency relationship between words in a sentence includes subject-predicate relationship, verb-object relationship, inter-object relationship, etc. The dependency syntactic relation of the sentence “evaluations play an invalid role in X.” is shown in Fig. 5. Among them, “play an invalid role in” is a formulaic language. From the figure, it can be seen that there are complex dependencies between these five words. Therefore, dependency syntactic parsing between words can express the dependency between two words. The closer the relationship is, the more likely it is to form a formulaic language.

**Feature representation based on GCN**

In the formulaic language identification using dependency syntactic relation, the existing studies mostly use the dependency syntax in the text to construct rules, extract features, and then input them into the classifier to recognize formulaic language through classification. Although this method can achieve certain results, the nonlinear semantic relationship between components in sentences has not been learned and utilized. Spatially, the relationship between words can be represented by a graph through mutual information and dependency parsing. The GCN is used to process the correlation information, so that the nonlinear semantic relationship between words in the sentence can be extracted.

In the feature representation based on GCN, the mutual information and dependency syntactic relation between words are used to determine the word connection relationship, and the basic features of words are represented as nodes. For the GCN with mutual information as input, firstly, the corpus is used as the data set to calculate the mutual information value between words, the word is used as the node, the mutual information value between nodes is used as the representation of the edge. The element \( a_{ij} \) in its adjacency matrix \( A \in \mathbb{R}^{N \times N} \) represents the mutual information value between the \( i \)th node and the \( j \)th node in the graph. For the GCN with dependency syntactic relation as input, firstly, the dependency syntax of sentences is analyzed, with words as nodes and the dependency between words as the representation of edges. The element \( a_{ij} \) in its adjacency matrix \( A \in \mathbb{R}^{N \times N} \) represents the dependency between the \( i \)th node and the \( j \)th node in the graph. If there is a dependency between the two nodes, the \( a_{ij} \) is 1, otherwise it is 0. For example, in the example of dependency syntactic parsing “evaluations play an invaluable
Figure 6  Adjacency matrix constructed based on dependency syntactic parsing.
Full-size DOI: 10.7717/peerjcs.984/fig-6

Figure 7  The structure diagram of GCN.
Full-size DOI: 10.7717/peerjcs.984/fig-7

role in X.”, the adjacency matrix A constructed based on dependency syntactic parsing is shown in Fig. 6.

In order to directly conduct deep learning modeling on graph data, the specific method adopts a variant model of convolution neural network-Graph Convolution Neural Network proposed by Kipf & Welling (2016), and the structure is shown in Fig. 7. The channels input by GCN is C, that is, the feature vector dimension of each node $X_i$ is C, the channels output by GCN is F, that is, the feature vector dimension of each node $Z_i$ is F, and finally the label $Y_i$ of the node is predicted by the feature of the node.

Specifically, given a graph $G = (V, E)$, V is a vertex set containing N nodes, E is an edge set including self-loop edges (that is, each vertex is connected to itself). The feature information of graph $G = (V, E)$ can be represented by Laplace matrix, as shown in
Eq. (11).
\[ L = D - A. \]  \hspace{1cm} (11)

Or use the symmetric normalized Laplace matrix, as shown in Eq. (12):
\[ L^{sy} = I_N - D^{-1/2}AD^{-1/2}. \]  \hspace{1cm} (12)

where \( A \) is the adjacency matrix of the graph, \( I_N \) is the \( n \)-order identity matrix; \( D = \text{diag}(d) \) is the degree matrix of the vertex, as shown in Eq. (13).
\[ D_{ii} = \sum_j A_{ij}. \]  \hspace{1cm} (13)

Based on the Fourier transform of graph, the graph convolution formula can be expressed as Eq. (14):
\[ g \ast x = U \left[ (U^T g) \cdot (U^T x) \right]. \]  \hspace{1cm} (14)

In the formula, \( x \) is the basic feature vector of the node; \( g \) is convolution kernel; \( U \) is the eigenvector matrix of Laplace matrix \( L \).

In order to reduce the amount of calculation, scholars used Chebyshev polynomials to simplify the graph convolution formula in 2017. Finally, the layered propagation formula of graph convolution can be expressed as Eq. (15):
\[ X^{(l+1)} = \sigma \left[ \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} X^{(l)} W^{(l)} \right]. \]  \hspace{1cm} (15)

In the formula, \( \tilde{A} = A + I_N \), \( \tilde{D} = \sum \tilde{A} \); \( \sigma \) is the activation function; \( W \) is the weight matrix to be trained.

**Label representation module**

Conditional Random Field (CRF) is a statistical-based learning model first proposed by *Lafferty, McCallum & Pereira (2001)*. Through the calculation and statistics of known features, the conditional probability of unknown features can be inferred. Based on the maximum entropy model and hidden Markov model, this model is mainly used for sequence tagging. It has achieved good recognition results in named entity recognition, syntactic analysis and part-of-speech tagging in the field of natural language processing.

In the model of this article, a conditional random field is added at the end of the model to predict the label category, so as to identify formulaic language in the sentence.

For an undirected graph \( G = (V, E) \), \( Y = Y_v | v \in V \) is a set of random variables \( Y_v \) indexed by node \( V \) in \( G \). For the conditions that \( X \) has been given, if each random variable \( Y_v \) obeys the Markov property, the probability of \( Y_v \) can be expressed as Eq. (16), then \((X, Y)\) is established as a conditional random field.
\[ P(Y_v | X, Y_u, u \neq v) = P(Y_v | X, Y_u, u \sim v). \]  \hspace{1cm} (16)

Among them, \( u \sim v \) represents that node \( u \) and node \( v \) are adjacent in the undirected graph \( G \).

When constructing CRF model, the linear-chain CRF commonly used in sequence labeling problem is selected. The structure diagram of linear-chain CRF is shown in Fig. 8.
Among them, the input observation sequence is represented by $X$, and $X$ is used as a precondition. $P(y|x)$ is defined as a joint probability distribution, and the maximum value among them is selected as the corresponding sequence label (Ren, Zhang & Zhang, 2019).

**EXPERIMENTAL SETUP AND ANALYSIS**

In order to recognize formulaic language in the text more accurately, this article proposes a formulaic language identification model based on GCN fusing associated information. It sets up three groups of comparative experiments. There are two purposes of the experiment. One is to select the optimal feature combination and complete the construction of the recognition model through ablation experiment and network layer comparison experiment. The second is to verify the effectiveness of the proposed model by comparing with the existing recognition models.

**Experimental data and labeling strategy**

We downloaded 20 articles in the computer field from the Web of Science, preprocessed the text, removed references, pictures, formulas, etc., and then segmented the sentences. A total of 1889 sentences were obtained as the data set. We take the formulaic language in the formulaic language corpus of Manchester University as the standard. Three professors of English major are invited to extract the formulaic language in the sentences. There are 2,563 formulaic languages in total, and 1,034 are left after the reprocessing. Then we tag the sentence. The tagging strategy adopts the BIO tagging method, ‘B’ means that the word is the starting position of the formulaic language, ‘I’ means that the word is the middle position of the formulaic language, and ‘O’ means that it does not belong to the formulaic language.

**Evaluating indicator**

In this article, the PRF index is used to evaluate the experimental results of formulaic language recognition. $P$ represents the accuracy of formulaic language recognition; $R$
represents the proportion of the correct number of recognized formulaic languages in the total number of formulaic languages in the corpus, which is called recall; F-score is the synthesis of P and R as the comprehensive index to evaluate the effect of formulaic language recognition. The three formulas correspond to the Eqs. (17), (18) and (19):

\[
P = \frac{N_m}{N_{total}} \times 100\% \tag{17}
\]

\[
R = \frac{N_m}{N_{correct}} \times 100\% \tag{18}
\]

\[
F = \frac{2 \times P \times R}{P + R}. \tag{19}
\]

Among them, \(N_m\) represents the number of correctly recognized formulaic language, \(N_{total}\) represents the total number of recognized formulaic language, and \(N_{correct}\) represents the total number of manually marked formulaic language.

**Parameter settings**

In the model, semantic features are 300-dimensional word vectors trained by Glove, part-of-speech features are 300-dimensional word vectors trained by the embedding layer. The basic feature extraction module uses Bi-LSTM to extract features, each layer has 128 neurons. The feature extraction module fusing associated information uses GCN to extract features, including two GCNs. The GCN based on mutual information has two layers of graph convolution, and the GCN based on dependency syntactic parsing has three layers of graph convolution. The output of the GCN layer is set to 64.

During model training, Adam optimizer is selected as the optimization algorithm, the learning rate is set to 0.001, and the decay ratio is set to 0.9.

**Ablation experiment**

In order to better verify the effect of formulaic language identification model based on GCN fusing associated information, seven comparative experiments were conducted to determine which feature is more important for formulaic language identification by setting ablation experiments. The specific methods are as follows:

- **Before_Bi-LSTM**: The part-of-speech features generated by PyTorch word embedding and the semantic features generated by GloVe word embedding are fused by early fusion. The fused feature vector is input into Bi-LSTM to extract the context semantic relationship. Finally, the features generated by Bi-LSTM are input into CRF to complete the recognition of formulaic language.

- **Before_CNN**: The part-of-speech features generated by PyTorch word embedding and the semantic features generated by GloVe word embedding are fused by early fusion. The fused feature vector is input into CNN. The size of convolution kernel is 3 * 3, and a total of two layers of CNN are set. Finally, the features generated by CNN are input into CRF to complete the recognition of formulaic language.
After_Bi-LSTM: The part-of-speech features generated by PyTorch word embedding and the semantic features generated by GloVe word embedding are fused by late fusion. That is, the two feature vectors are input into Bi-LSTM respectively, and then the processed vectors are fused. Finally, the fused features are input into CRF to complete the recognition of formulaic language.

After_Bi-LSTM_CNN: Based on After_Bi-LSTM, this method adds a layer of CNN between Bi-LSTM and CRF. The size of convolution kernel is 3 * 3, and a total of two layers of CNN are set.

Bi-LSTM_DS_GCN: The part-of-speech features generated by PyTorch word embedding and the semantic features generated by GloVe word embedding are fused by late fusion. The fused feature vector is used as the basic feature and the matrix generated by dependency syntactic relation is used as the associated feature, which are input into GCN for feature representation. Finally, the features generated by GCN are input into CRF to complete the recognition of formulaic language.

Bi-LSTM_MI_GCN: The difference from Bi-LSTM_DS_GCN is that the matrix generated by dependency syntactic relation is replaced with the matrix generated by the mutual information between words as the associated feature.

Bi-LSTM_MI_DS_GCN: This is the model proposed in this article. The late fused part-of-speech features and semantic features are used as the basic features and the matrices generated according to mutual information and dependency syntactic relation are used as the correlation information. The basic features and correlation information are input into GCN. Finally, the features are decoded through CRF.

Seven methods are tested on the data set, and the experimental results are shown in Table 2 and Fig. 9. In Table 2, train_time represents the time required to train 100 pieces of data, and test_time represents the time required to test 100 pieces of data.

Analysis of experimental results:

(1) The difference between Experiment 1 and Experiment 2 lies in the comparison between Bi-LSTM and CNN. The experimental results show that Bi-LSTM is much better than CNN in feature extraction. Because the identification of formulaic language is a typical sequence labeling problem, Bi-LSTM can capture the long-distance and long-term dependence of sentence context information and can capture the two-way information of sentences. However, CNN cannot capture the long-distance dependence information well, so it is better to use Bi-LSTM in formulaic language recognition. It is worth noting that the Recall in the result of using CNN to extract features is relatively high, indicating that it can recognize more formulaic languages. However, it can also recognize many non-formulaic languages, so the Precision is not very high.

(2) The difference between Experiment 1 and Experiment 3 is that the way of feature fusion is different. Experiment 1 adopts early fusion and Experiment 3 adopts late fusion. The experimental results show that the late fusion method has higher Precision and the early fusion method has higher Recall. The main reason is that the early fusion method can recognize more results when recognizing formulaic languages, but it can also recognize many non-formulaic languages, so its Precision is low. The features of the late fusion
### Table 2  Comparison of experimental results of ablation experiment.

| Number | Method               | Precision | Recall | F1-score | Train_time | Test_time |
|--------|---------------------|-----------|--------|----------|------------|-----------|
| 1      | Before_Bi-LSTM      | 62.12     | 81.46  | 70.49    | 9.40775    | 3.83115   |
| 2      | Before_CNN          | 34.23     | 50.33  | 40.75    | 11.24228   | 5.06429   |
| 3      | After_Bi-LSTM       | 77.08     | 73.51  | 75.25    | 11.96498   | 5.16604   |
| 4      | After_Bi-LSTM_CNN   | 72.46     | 66.23  | 69.20    | 22.86475   | 8.34037   |
| 5      | Bi-LSTM_DS_GCN      | 65.22     | 69.54  | 67.31    | 12.20419   | 4.36681   |
| 6      | Bi-LSTM_MI_GCN      | 74.19     | 76.16  | 75.16    | 11.19024   | 3.63964   |
| 7      | Bi-LSTM_MI_DS_GCN   | **81.71** | **85.37** | **83.50** | 13.55199   | 5.35704   |

![Graph showing comparison of experimental results](Full-size DOI: 10.7717/peerjcs.984/fig-9)

method are more accurate, and the recognition results are not as much as those of the early fusion, but the formulaic languages can be recognized more accurately. According to the F1-scores of the two methods, the effect of late fusion is better than that of early fusion.

(3) Experiment 4 adds CNN layer based on Experiment 3, but the results after adding CNN are not as good as the previous results. The main reason is that CNN captures local correlations and each layer of CNN has a fixed span. Naturally, this layer can only model limited distance information. In Experiment 4, Bi-LSTM has obtained the long-distance dependency of the context, and then CNN is added after Bi-LSTM to deeply abstract the features, some correct formulaic languages are filtered out, resulting in a significant decline in Precision and Recall.

(4) Experiment 5 adds feature extraction based on GCN fusing dependency syntactic parsing on the basis of Experiment 3. The experimental results showed that after adding dependency syntactic features, the Recall and Precision are reduced. Because dependency syntactic relation mainly focuses on the dependency between two words in a sentence, it is easy to cause that the extracted word string does not belong to formulaic language, so the recognition effect is not good.
Experiment 6 adds feature extraction based on GCN fusing mutual information on the basis of Experiment 3. The experimental results show that after adding the mutual information features, the Recall increases, indicating that the number of correctly recognized formulaic language is more. However, due to the reduction of Precision, F1-score is not improved. At the same time, compared with experiment 5, it shows that mutual information features can better represent the characteristics of formulaic language than dependency syntactic features.

(5) Experiment 7 (the model proposed in this article) inputs the dependency syntactic features and mutual information features into GCN for feature extraction. The experimental results show that although the independent dependency syntax features (Experiment 5) do not show good results, after combining the two, the Precision, Recall have increased significantly. The reason is that the features of dependency syntactic parsing and mutual information complement each other in the identification of formulaic language to achieve efficient extraction of formulaic language. It also shows that dependency syntax features and mutual information features are important features to measure formulaic language.

(6) From the train_time and test_time in Table 2, it can be seen that even though the number of features fused by the model in this article increases, the training time does not increase much, and the training time is less than the method using CNN. At the same time, the time for the model to test 100 pieces of data is only 3.7% higher than that of experiment 3 (the experiment with the highest F1-score among other experiments). This shows that the model in this article does not reduce the efficiency of formulaic language identification while improving the Precision and Recall. It is an efficient and accurate model for the identification of formulaic language.

In summary, through ablation experiments, the results verify the effect of formulaic language identification model based on GCN fusing associated information. That is, through the late fused part-of-speech features and semantic features as the basic features, dependency syntactic relation and mutual information as the associated information, the combined model can reduce the errors of a single model and enhance their advantages.

**Comparative experiment of different network layers**

Because the model in this article involves two GCNs, one is GCN based on dependency syntactic parsing, and the other is GCN based on mutual information. Therefore, when selecting the number of layers of GCN, we use Experiment 5 and Experiment 6 in ablation experiment to do two groups of comparative experiments and select the optimal number of network layers by comparing the experimental results.

(1) Experiment 5 is a GCN structure based on dependency syntactic parsing. Experiments are carried out by setting 1, 2, 3, 4 and 5-layer graph convolution respectively. The experimental results are shown in Fig. 10. As can be seen from the figure, the effect of using 3-layer graph convolution is the best for dependency syntactic parsing.

(2) Experiment 6 is a GCN structure based on mutual information. Experiments are carried out by setting 1, 2, 3, 4 and 5-layer graph convolution respectively. The experimental
The results are shown in Fig. 11. As can be seen from the figure, the effect of using 2-layer graph convolution is the best for mutual information.

Analysis of experimental results:

It is generally believed that increasing the number of network layers can reduce the error and improve the accuracy, but it will also complicate the network, thus increasing the training time of the network and the tendency of over fitting. Therefore, the number
of network layers required by different features is different. In this article, dependency syntactic parsing only uses 0 or 1 to describe the relationship between two words, which is not good enough for the interpretation of formulaic language, so it needs three-layer graph convolution to extract better features. Mutual information can better represent the characteristics of formulaic language, so two-layer graph convolution can extract better features.

From the above analysis, it can be seen that in formulaic language identification model based on GCN fusing associated information, the layers of the two GCNs are set to 2 and 3 layers respectively.

**Comparative experiments of different models**

In order to verify the effectiveness of the model proposed in this article, the CNN_Bi-LSTM_CRF model in literature (Gao, Peng & Bai, 2020) and the Bi-LSTM_CRF model in literature (Berk, Erden & Güngör, 2018) are selected for comparison.

- **CNN_Bi-LSTM_CRF:** This article performs the task of named entity recognition. Because formulaic language recognition and named entity recognition are similar, and this model performs well in the field of named entity recognition, this model is used as a comparative experiment. The word vector is trained with word2vec, and the word vector of the text data obtained after word2vec training is spliced to generate the word vector matrix, which is then used as the input of CNN convolution layer. The CNN module extracts the spatial feature information of the text through convolution and the collection of vector matrix. Then, the results are input into Bi-LSTM for forward and backward training. Finally, the vector with sentence feature information is put into CRF for decoding and prediction to obtain the final sequence.

- **Bi-LSTM_CRF:** This article describes the Deep-BGT system that participated in the PARSEME shared task 2018 to automatic identification of verbal multi-word expressions (VMWEs). Author uses Bi-LSTM model with a CRF layer on top. The input layer includes word vectors generated by fastText word embedding technology, POS and dependencies. Each input vector is represented as splicing these three features, similar to early fusion technology. Because formulaic language is also a kind of multi-word expression, the combination of Bi-LSTM and CRF is the mainstream method in the field of multi-word expression recognition, so this model is used as a comparative experiment.

The experimental results of the above two models and the model proposed in this article on formulaic language recognition are shown in Fig. 12.

Analysis of experimental results:

1. As the CNN_Bi-LSTM_CRF model is the primary method in named entity recognition, the experimental results show that this model does not perform well in the task of formulaic language recognition. It can be seen that in different tasks, although the two tasks are very similar, we should also start from the essence of the object in the task, mine the characteristics that can represent the research object, and design a unique model. At the same time, through the CNN_Bi-LSTM_CRF model and Experiments 2 and 4 in
the previous section, it can be found that the effect will not get better after adding CNN to the model. Hence, CNN is not suitable for formulaic language identification task.

(2) Bi-LSTM_CRF model is used to identify multi-word expressions. Compared with Experiment 1 in the previous section, the difference lies in the different input features. The input features of Experiment 1 are part-of-speech features and GloVe word embedding features, while Bi-LSTM_CRF model inputs part-of-speech features, fastText word embedding features and dependency syntactic relation. The experimental results show that the F1-score of Bi-LSTM_CRF model is higher. However, the Recall of Experiment 1 is higher, indicating that dependency syntactic relation is conducive to recognizing formulaic language.

At the same time, in comparison with the model in this article, the main difference is that the Bi-LSTM_CRF model only constructs the dependency syntactic relation into a simple feature vector, splices it with other features and then trains them. The model in this article constructs the graph structure through the syntactic dependency tree and then extracts the features through GCN. The advantage of GCN is that it can gather the information on all edges and points. Thus, the boundary ambiguity between words is eliminated, and any two non-adjacent nodes in the graph are each other’s second-order neighbors. They can receive each other’s non-local information through two-node updates. The aggregated features in this method can more accurately represent formulaic language, so the effect of identifying formulaic language is better.

**ERROR ANALYSIS**

In order to make an in-depth analysis of the shortcomings of this model, we investigated the examples of formulaic language recognition errors under the best experimental results and selected three representative sentences for analysis. The specific contents are shown in Table 3.

It can be seen from the table that the reason for the recognition error in the first example is that the word ‘since’ is not recognized. The reason for the recognition error in the second
example is that the word ‘to’ is recognized. The reason for the recognition error in the third example is that ‘restore to’ is also regarded as a formulaic language. For the first two examples, in fact, the model’s positioning in the sentence is very accurate. However, due to the uncertainty of manual labeling, fixed evaluation indicators and other factors, it can only be considered that this word string recognition is wrong. If the recognition effect is viewed from the perspective of word coverage, it must be higher than the existing evaluation indicators.

In the third example, although ‘restore to’ is a non-formulaic language, it contains many characteristics of formulaic language, resulting in the model mistakenly considering it as formulaic language. This kind of word string usually contains many characteristics of formulaic language, such as Verb + preposition and other structures. However, it may not have a specific function and meaning in the fields of scientific and technological literature writing, so they are not formulaic languages. This situation is a problem we should think about and solve in the future.

**CONCLUSIONS**

In this article, a formulaic language identification model based on GCN fusing associated information is proposed. The part-of-speech features and semantic features fused through late fusion are taken as the basic features. Then the two associated information of mutual information and dependency syntactic relation are input into GCN together with the basic features for feature representation. This combined representation can capture the syntactic and semantic structure of multi semantic web. It can also conduct a more in-depth downstream semantic analysis. Finally, we input the feature vector output from GCN layer to CRF layer for decoding, obtain the label category of each character, and get the formulaic language. Several groups of comparative experiments show that compared with the existing models, the proposed model can improve the accuracy of formulaic language identification and verify the model’s effectiveness. In addition, it should be noted that the formulaic language identification model proposed in this article can obtain strong identification performance with only a small proportion of labeled text. For the identification of other tasks, it has a certain universal reference value. However, this model still has some shortcomings in analyzing the fuzziness and fuzzy boundary of some
formulaic languages. The word strings described by facts in sentences cannot be understood and learned by the model, which needs further consideration in future research.

**ADDITIONAL INFORMATION AND DECLARATIONS**

**Funding**
This article is supported by the 2020 Jilin Provincial Social Science Fund Project "Recognition and Application of English and Chinese Academic Formulaic Language" (No. 2020B206), the National Key R&D Program of China ”The Framework Design and Verification of Data Space Management Engine and Management Service Based on Multi-value Chain Collaboration” (No. 2020YFB1707804), and the Jilin City Science and Technology Innovation Development Project ”Study on the Emotion Classification of Jilin Tourism Online Review Text” (No. 20200104108). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Grant Disclosures**
The following grant information was disclosed by the authors:
- 2020 Jilin Provincial Social Science Fund Project: 2020B206.
- National Key R&D Program of China: 2020YFB1707804.
- Jilin City Science and Technology Innovation Development Project: 20200104108.

**Competing Interests**
The authors declare there are no competing interests.

**Author Contributions**
- Fanqi Meng performed the computation work, authored or reviewed drafts of the article, and approved the final draft.
- Yujie Zheng conceived and designed the experiments, performed the experiments, prepared figures and/or tables, and approved the final draft.
- Songbin Bao analyzed the data, authored or reviewed drafts of the article, and approved the final draft.
- Jingdong Wang conceived and designed the experiments, authored or reviewed drafts of the article, and approved the final draft.
- Shuaisong Yang analyzed the data, prepared figures and/or tables, and approved the final draft.

**Data Availability**
The following information was supplied regarding data availability:

The code and data is available in the Supplemental Files and at Github: https://github.com/access1-sys/Syntax_analysis.

**Supplemental Information**
Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.984#supplemental-information.
REFERENCES

Abbas MA, Hammad S, Hwang GJ, Khan S, Gilani SM. 2020. An assistive environment for EAL academic writing using formulaic sequences classification. Interactive Learning Environments 28:1–15.

Ashok A, Elmasri R, Natarajan G. 2019. Comparing different word embeddings for multiword expression identification. In: International conference on applications of natural language to information systems. Cham: Springer, 295–302.

Berk G, Erden B, Güngör T. 2018. Deep-bgt at parseme shared task 2018: bidirectional lstm-crf model for verbal multiword expression identification. In: Proceedings of the joint workshop on linguistic annotation, multiword expressions and constructions (LAW-MWE-CxG-2018). Santa Fe 248–253.

Chen YH, Baker P. 2010. Lexical bundles in L1 and L2 academic writing. Language Learning & Technology 14(2):30–49.

Durrant P, Mathews-Aydnl J. 2011. A function-first approach to identifying formulaic language in academic writing. English for Specific Purposes 30(1):58–72 DOI 10.1016/j.esp.2010.05.002.

Gao D, Peng L, Bai Y. 2020. HAZOP Text Named Entity Recognition using CNN-BiLSTM-CRF model. In: 2020 Chinese Automation Congress (CAC). Shanghai 6159–6164.

Gharbieh W, Bhavsar V, Cook P. 2017. Deep learning models for multiword expression identification. In: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017). Vancouver 54–64.

Hochreiter S, Schmidhuber J. 1997. Long short-term memory. Neural Computation 9(8):1735–1780 DOI 10.1162/neco.1997.9.8.1735.

Hu XR, Yao CQ, Gao YF. 2019. Phrase recognition method combining multiple strategies. Information Science 37(6):49–54.

Hyland K. 2008. Academic clusters: text patterning in published and postgraduate writing. International Journal of Applied Linguistics 18:41–62 DOI 10.1111/j.1473-4192.2008.00178.x.

Iwatsuki K, Boudin F, Aizawa A. 2020. Extraction and evaluation of formulaic expressions used in scholarly papers. Language Learning 67:885–992.

Khaleghi B, Khamis A, Karray F. 2013. Multisensor data fusion: a review of the state-of-the-art. Information Fusion 14(1):28–44 DOI 10.1016/j.inffus.2011.08.001.

Kipf TN, Welling M. 2016. Semi-supervised classification with graph convolutional networks. ArXiv preprint. arXiv:1609.02907.

Kirilin A, Krauss F, Versley Y. 2016. ICL-HD at SemEval-2016 Task 10: improving the detection of minimal semantic units and their meanings with an ontology and word embeddings. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). San Diego 937–945.

Klyueva N, Doucet A, Straka M. 2017. Neural networks for multi-word expression detection. In: Proceedings of the 13th workshop on multiword expressions (MWE 2017). Valencia 60–65.
Lafferty J, McCallum A, Pereira FCN. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data.

Lei L, Liu D. 2018. The academic English collocation list: a corpus-driven study. *International Journal of Corpus Linguistics* 23(2):216–243 DOI 10.1075/ijcl.16135.lei.

Li GC. 2012. An overview of foreign research on formulaic language identification. *Journal of Central South University* 18(5):221–225.

Meng FQ, Cheng WY, Wang JD. 2021. Semi-supervised software defect prediction model based on tri-training. *KSII Transactions on Internet and Information Systems* 15(11):4028–4042.

Meng FQ, Yang SS, Wang JD, Xia L, Liu H. 2022. Creating knowledge graph of electric power equipment faults based on BERT–BiLSTM–CRF model. *Journal of Electrical Engineering & Technology* 17:1–10.

Metin SK. 2018. Feature selection in multiword expression recognition. *Expert Systems with Applications* 92:106–123 DOI 10.1016/j.eswa.2017.09.047.

Moreau E, Alsulaimani A, Maldonado A. 2018. Crf-seq and crf-deptree at parseme shared task 2018: Detecting verbal mwes using sequential and dependency-based approaches. In: *Joint workshop on linguistic annotation, multiword expressions and constructions (LAW-MWE-CxG-2018) at the 27th International Conference on Computational Linguistics (COLING 2018)*. Santa Fe 241–247.

Pecina P. 2009. Lexical association measures: collocation extraction. In: *Volume 4 of studies in computational and theoretical linguistics*. Praha, Czechia: UFAL.

Pennington J, Socher R, Manning CD. 2014. Glove: global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Doha 1532–1543.

Ren M, Zhang Q, Zhang J. 2019. An introductory survey of probability density function control. *Systems Science & Control Engineering* 7(1):158–170 DOI 10.1080/21642583.2019.1588804.

Simpson-Vlach R, Ellis NC. 2010. An academic formulas list: new methods in phraseology research. *Applied Linguistics* 31(4):487–512 DOI 10.1093/applin/amp058.

Strobl C, Ailhaud E, Benetos K, Devitt A, Kruse O, Proske A, Rapp C. 2019. Digital support for academic writing: a review of technologies and pedagogies. *Computers & Education* 131:33–48 DOI 10.1016/j.compedu.2018.12.005.

Taslimipoor S, Bahaadini S, Kochmar E. 2020. MTLB-STRUCT@PARSEME 2020: capturing unseen multiword expressions using multi-task learning and pre-trained masked language models. arXiv preprint.

Taslimipoor S, Rohanian O. 2019. Cross-lingual transfer learning and multitask learning for capturing multiword expressions. In: *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*. Florence 155–161.

Wahl A. 2019. Computational extraction of formulaic sequences from corpora: two case studies of a new extraction algorithm. *International Journal of Corpus Linguistics* 33(5):752–793.

Yu LX. 2019. Corpus-based formulaic language research at home and abroad-review and summary. *Overseas English* 5(2):221–222.
Zhang Q, Wang A. 2016. Decoupling control in statistical sense: minimised mutual information algorithm. *International Journal of Advanced Mechatronic Systems* 7(2):61–70 DOI 10.1504/IJAMECHS.2016.082625.