Journal of Computers Vol. 33 No. 3, June 2022, pp. 35-47
doi: 10.53106/199115992022063303003

Prediction of Academic Formulaic Language based on Multi-feature Fusion

Fanqi Meng[1, 2], Yujie Zheng[1], Jingdong Wang[1*], Songbin Bao[3]

[1] School of Computer Science, Northeast Electric Power University, Jilin 132012, China
{249925066, 278764886, 286604374}@qq.com
[2] School of Information Engineering, Guangdong Atv Academy For Performing Arts, Guangdong 523710, China
[3] School of Foreign Language, Northeast Electric Power University, Jilin 132012, China
Baosongbin@163.com

Received 16 June 2021; Revised 28 September 2021; Accepted 13 December 2021

Abstract. Academic formulaic language is multi-word combinations with specific functions and semantics, which are important to improve the idiomaticity, fluency and logic of machine translation, intelligent question answering, automatic summarization, etc. In order to narrow the search range of the corpus and extract academic formulaic language more efficiently, this paper proposes a prediction model of academic formulaic language based on multi-feature fusion. The semantic features and part-of-speech features of the academic formulaic language are extracted separately, and then the late fusion method is used to learn multiple features and predict whether formulaic language is included in the sentence. Experimental results show that the late fusion method based on part-of-speech features and semantic features has the best predictive effect among the four fusion methods, which lays the foundation for further efficient recognition of academic formulaic language.

Keywords: academic formulaic language, multi-features, late fusion

1 Introduction

Academic formulaic language is a kind of language block with a high frequency. It usually has a relatively complete structure, meaning and function and it is generally recognized, stored and extracted as a whole form [1]. According to statistics from different researchers, 40%-60% of spoken or written English is composed of this kind of patterned chunks [2], especially in the writing of scientific and technological literature, the use of appropriate and natural academic formulaic language plays an important role. In the practical application of natural language processing (NLP), academic formulaic language is conducive to the idiomaticity, fluency and logic of machine translation, intelligent question answering, automatic summarization [3].

At present, the research on academic formulaic language at home and abroad is either corpus based descriptive research or confirmatory research for language acquisition. There are very few researches related to the prediction and recognition of formulaic language using computer technology [4]. In fact, the prediction and recognition of formulaic language is the basis of the application research of formulaic language. Early methods of manually extracting academic formulaic language consumed a lot of manpower and material resources, so the traditional manual extraction method of academic formulaic language is no longer applicable. In recent years, academic formulaic language recognition methods mainly include statistical-based methods, rule-based methods, and machine learning methods. However, if the number of corpora is huge, the semantics of the text is diverse and the content is complex, the efficiency and accuracy of directly using these methods to identify academic formulaic language are very low. Therefore, we first predict the text and judge whether the sentence contains academic formulaic language, so as to narrow the data range to be input in the task of academic formulaic language recognition and improve the recognition efficiency. Based on this, this article proposes an academic formulaic language prediction model based on multi-feature fusion. The main technical contributions are summarized as follows:

(1) This paper extracts the features that can represent academic formulaic language: using the embedding layer in PyTorch to generate word embedding vectors as part-of-speech features, and using the feature vectors trained by GloVe word vector technology as semantic features. Through experiments, it is found that these two features are very important for academic formulaic language.

(2) The early fusion and late fusion technologies are proposed. By comparing the two feature fusion methods, it is found that the late fusion of features has a better effect on the representation of features.

(3) A classification model for judging whether the sentence contains academic formulaic language is proposed,

* Corresponding Author

and then this model is used in the prediction task. An academic formulaic language prediction model based on multi-feature fusion is proposed, which can screen the sentences with high probability containing academic formulaic language from large-scale corpus, reduce the corpus range and improve the efficiency of academic formulaic language recognition task.

2 Related Work

The Academic Phrasebank website[1] created by Dr. John Morley of the University of Manchester, all the phrases in it are from real academic materials. These phrases were originally extracted from 100 postgraduate papers of the University of Manchester, aiming to provide students with some common sentence patterns or phrase resources in academic paper writing. It can help students organize language and content effectively, but such phrase library lacks contextual information and the content is not targeted [5]. What's more, it is necessary to manually extract the academic phrases in the papers to update the phrase database, which consumes a lot of manpower and material resources. Therefore, the traditional artificial-based formulaic language identification method is no longer applicable. For this reason, many scholars at home and abroad have carried out a series of research work on the task of formulaic language identification.

At present, formulaic language identification mainly adopts statistics-based method, rule-based method or machine learning method.

Statistics-based recognition method counts phrase features from the text, such as word frequency, mutual information, information entropy, etc. [6]. Pecina proposed a method of association measurement, which considers the phrase frequency and also considers other phrases with strong relevance to the phrase to determine whether it can be combined into a formulaic language [7]. On this basis, Alexander Wahl proposed the MERGE (Multiword Expressions from the Recursive Grouping of Elements) model, which based on the strength of vocabulary association, iteratively combines adjacent phrases into gradually longer sequences [8]. With the development of N-gram, Hyland, Chen, Simpson-Vlach proposed to extract N-grams from the corpus, but the simple N-gram model has the problem of too large parameter space and severe data sparseness [9-11], so O'Donnell proposed an adjusted frequency list (AFL) [12]. First, all N-grams in the corpus that reach the threshold are retrieved, then only the N-grams and their frequencies that exceed the frequency threshold are retained in AFL, and finally they are sorted according to frequency. The main defect of statistical based recognition method is that it only considers word frequency and relevance, and does not consider the meaning and function of formulaic language. Therefore, it is easy to ignore some formulaic language composed of low-frequency words with clear textual function, but extract too many formulaic language composed of high-frequency words without clear textual function.

The rule-based recognition method is to artificially establish some part-of-speech and syntactic rules to extract formulaic language [13]. Iwatsuki believes that a sentence is composed of a formulaic part and an unformulated part, so he proposed to use named entities and dependent structures to remove the unformulated part of the sentence, and the rest is the formulaic part [14]. Liu proposed the LDA-Based Sequential Labelling model, which used the topic model to remove unnecessary words from sentences to achieve the purpose of identifying formulaic language [15]. At the same time, some researchers have tried to identify formulaic language through string matching and regular expressions [16]. Through regression analysis, it is found that the number of rules is positively correlated with the types of recognized formulaic language. The more rules there are, the more formulaic language is recognized, but the defined rules can't cover all formulaic language, and this method does not consider the characteristics of word frequency, so it is not flexible and comprehensive.

With the development of machine learning, Hidden Markov Model (HMM), Conditional Random Field (CRF), Maximum Entropy (ME) and Decision Tree (DT) have been used to identify formulaic language, Abbas tried to use classifiers such as Naive Bayes Model (NBM) and Logistic Regression (LR), and found that the recognition results were not good, and finally chose to use RF and SVM classifiers to recognize formulaic language through classification [17]. Gharbieh first tried to use CNN to recognize formulaic language [18]. They used word2vec training feature vectors. Ashok 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 [19]. For the recognition method of machine learning, it is necessary to extract the features that can represent the samples. The more appropriate the feature selection, the higher the accuracy of recognition, so feature selection is an urgent problem to be solved [20].

To sum up, in view of the low efficiency and accuracy of academic formulaic language recognition, from the perspective of narrowing the search range of the corpus, this article proposes a method to use a trained model to

1 http://www.phrasebank.manchester.ac.uk/

predict whether a sentence contains academic formulaic language. It filters out sentences with a high probability of containing academic formulaic language, which greatly reduces the corpus of academic formulaic language recognition tasks, and can improve the efficiency and accuracy of recognition. In order to more comprehensively apply the part-of-speech and semantic features of sentences, this paper proposes an academic formulaic language prediction model based on multi-feature fusion. It uses the embedding layer in PyTorch to generate word embedding vectors as part-of-speech features, uses the feature vectors trained by GloVe word vector technology as semantic features. The Bi-LSTM model is used to learn the two features, and the learned features are fused through the late fusion method, finally it is predicted whether the sentence contains academic formulaic language.

3 Predictive Model of Academic Formulaic Language

3.1 Overall Framework

The overall structure of the model in this paper is mainly divided into two parts: the training phase and the prediction phase. The overall framework is shown in Fig. 1. First, we preprocess the formulaic language in the academic phrase library to obtain more structured data, and then use the Stanford CoreNLP to tag the processed data. The features of the input model include semantic features and part-of-speech features. The semantic features are 300-dimensional word vectors pre-trained by GloVe. For part-of-speech features, the embed-ding layer in PyTorch is used for training to generate 300-dimensional word embedding vectors. Through two different fusion methods, the feature vector is input into Bi-LSTM to train the model. We extract 50 papers in the computer field from the top journals, and preprocess the entire paper, then extract features, finally input them into the trained model. Finally, the model predicts the probability of each sentence containing academic formulaic language. If the probability is greater than the threshold, the sentence will be accepted and stored in a corpus; otherwise, it will be deleted.

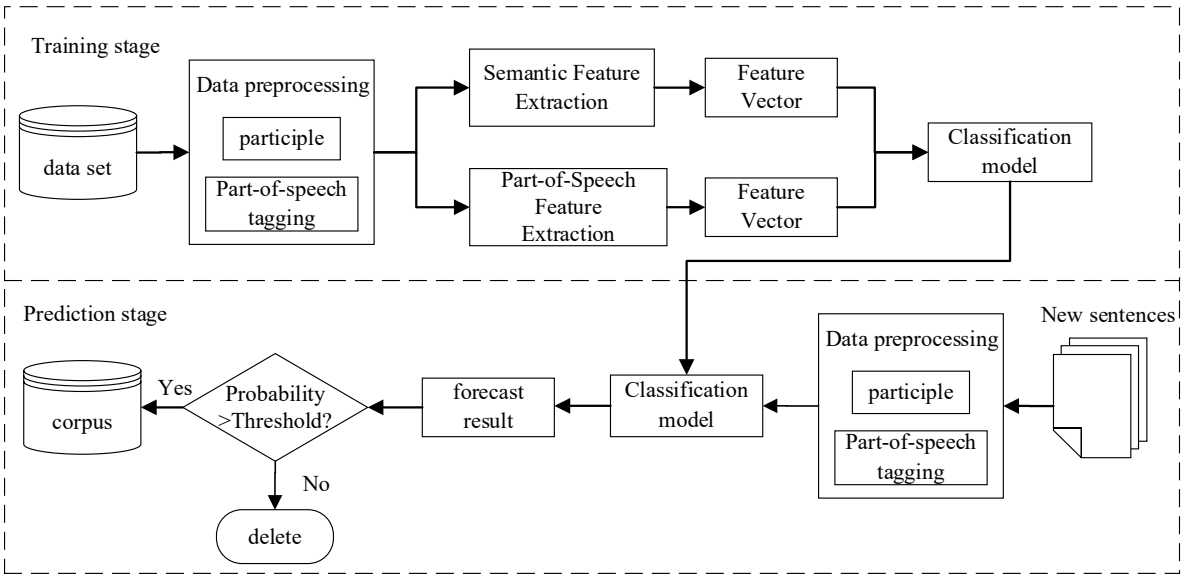

Fig. 1. The overall framework of formulaic language prediction

3.2 Feature Extraction

In the process of NLP, computer cannot directly use the text data, the text data needs to be expressed as a feature vector, and then the feature vector is used as the input of the model [21]. In this paper, the embedding vector generated by the embedding layer in PyTorch is used as part-of-speech feature, the feature vector trained by GloVe word vector technology is used as semantic feature. Through different fusion methods, the feature vectors are fused and then used as the feature of the recognition academic formulaic language.

3.2.1 Part-of-speech Feature Extraction

The biggest difference between academic formulaic language and general formulaic language is that the forms of academic formulaic language are mostly fixed. For example, Subject-Link verb-Predicative Structure or Subject Verb Object is easier to be an academic formulaic language, so the part-of-speech feature is used as one of the characteristics of the recognition formulaic language. In this article, we first use the Stanford part-of-speech tagger to perform part-of-speech analysis on the preprocessed sentences. Examples of the results of part-of-speech analysis 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 academic formulaic language.

Table 1. Examples of the results of part-of-speech analysis

Academic 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

For each result after Stanford part-of-speech tagging, a unique code is assigned (the part-of-speech coding is shown in Table 2), so that the text data is converted into a vector. We input the vector into the embedding layer for training, and a word embedding vector is generated as a part-of-speech feature.

Table 2. Part-of-speech coding correspondence table

PAD	Fill character	0	MD	modal auxiliary	12	RP	particle	24
UNK	Special characters	1	NN	singular or mass	13	TO	"to"	25
CC	conjunction	2	NNP	proper, singular	14	VB	base form	26
CD	Cardinal	3	NNPS	noun, proper	15	VBD	past tense	27
DT	determiner	4	NNS	common	16	VBG	gerund	28
EX	Existential sentence	5	PDT	pre-determiner	17	VBN	Past participle	29
FW	foreign word	6	POS	genitive marker	18	VBP	present tense(n)	30
IN	Preposition	7	PRP	pronoun, personal	19	VBZ	present tense(y)	31
JJ	Adjective, numeral	8	PRP$	possessive	20	WDT	WH-determiner	32
JJR	Comparative adjective	9	RB	adverb	21	WP	WH-pronoun	33
JJS	Superlative adjective	10	RBR	Comparative adverbs	22	WP$	possessive	34
LS	list item marker	11	RBS	Superlative adverb	23	WRB	Wh-adverb	35

3.2.2 Semantic Feature Extraction

For the processing of text, the characteristic items such as characters, words, and phrases are the main objects of processing, but the characters, words, and phrases more reflect the vocabulary information of the text, rather than its semantic information, therefore they cannot accurately express the content of the text. For academic formulaic language, it is a multi-word unit with a high frequency of occurrence, with a relatively complete structure, meaning and function. And one of the words can be replaced with words with similar meanings, so semantic features are important features of academic formulaic language and the key to identifying formulaic language. This paper uses GloVe to extract the features of the text, and the 300-dimensional feature vector obtained is used as the semantic feature of the input model.

The full name of GloVe is Global Vectors for Word Representation. It is a word representation tool based on

global word frequency statistics. It can express words as a vector composed of real numbers. These vectors capture some semantic characteristics between words, such as similarity [22]. The core idea of GloVe word vector is to use the number of co-occurrences between words for training. The so-called co-occurrence is the number of simultaneous appearances of two words within a context window. The context window is traversed once from the beginning to the end of the corpus. A global co-occurrence matrix is obtained, and then word vectors are trained based on this co-occurrence matrix. At the same time, cosine similarity is used to measure the similarity between vectors, and words with similar semantics can be found. In summary, the feature vectors trained by GloVe can fully express the semantic features of academic formulaic language.

3.3 Bi-LSTM Model

Recurrent Neural Network (RNN) was proposed by Elman in 1990 [23]. This model can process sequence data and input the data into the model one by one according to the time series. However, the traditional RNN network cannot use the future information of the sentence, cannot model the long-distance information well, and is prone to problems such as gradient disappearance and gradient explosion. To solve the above problems, Hochreiter proposed a Long Short-Term Memory (LSTM) model in 1997 [24]. Based on the RNN unit, LSTM adds a cell state to record the information passed over time. Through the input gate, output gate and forget gate to realize the selective utilization of historical information, the information of longer sequence data can be captured effectively.

Let $X=[x_1, x_2, ..., x_t]$ be the input text, the realization of the internal structure of LSTM neuron is as follows:

$$i_t=\sigma(W_{xi}x_t+W_{hi}h_{t-1}+W_{ci}c_{t-1}+b_i). \tag{1}$$

$$f_t=\sigma\left(W_{xf}x_t+W_{hf}h_{t-1}+W_{cf}c_{t-1}+b_f\right). \tag{2}$$

$$c_t=f_tc_{t-1}+i_t\tanh\left(W_{xc}x_t+W_{hc}h_{t-1}+b_c\right). \tag{3}$$

$$o_t=\sigma(W_{xo}x_t+W_{ho}h_{t-1}+W_{co}c_t+b_o). \tag{4}$$

$$h_t=o_t\tanh\left(c_t\right). \tag{5}$$

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; σ is the Sigmoid function; *tanh* is the hyperbolic tangent function.

The Bi-LSTM learns sequence data from left to right and from right to left through two layers of LSTM neurons. The structure of Bi-LSTM is shown in Fig. 2.

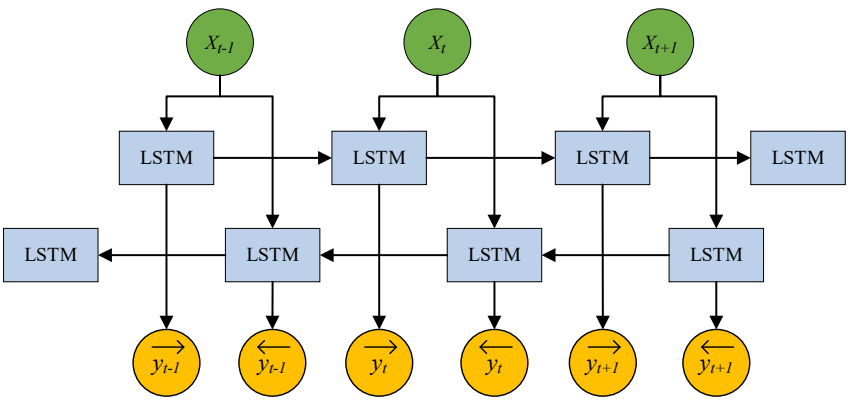

Fig. 2. The structure diagram of Bi-LSTM

In Fig. 2, x_t represents the input of the network at time t, the LSTM in the box is the standard LSTM model,

$\overrightarrow{y_t}$ is the output of the forward LSTM at time t, and $\overleftarrow{y_t}$ is the output of the reverse LSTM at time t. The output representation of Bi-LSTM at time t is defined as $y_t=[\overrightarrow{y_t} : \overleftarrow{y_t}]$, that is, the output at time t is directly spliced by the forward output and the reverse output.

3.4 Feature Fusion

In many works, a fusion of different features is an important mean to improve the performance of the model. Low-level features contain more detailed information, but due to less processing, they have lower semantics and more noise. High-level features have stronger semantic information, but their ability to perceive details is poor [25]. How to efficiently integrate the two, taking the strong points and discarding the bad ones, is the key to improving the model. This paper uses two fusion methods (early fusion and late fusion) to fuse the extracted features, and compares the performance of the two fusion methods through experiments.

3.4.1 Early Fusion of POS Features and Semantic Features

Early fusion is to fuse multiple layers of features first, and then train the model on the fused features (only after the complete fusion, the training will be carried out uniformly). There are two classic early fusion methods:

(1) Series feature fusion, which directly connects two features. If the dimensions of the two input features x and y are p and q, the dimension of the output feature z is $p + q$;

(2) Parallel strategy, which combines these two feature vectors into a complex vector, for the input features x and y, $z = x + iy$, where i is the imaginary unit [26].

The early fusion of this article uses the first method, that is, the part-of-speech feature vector and semantic feature vector are simply spliced, and the combined features are input into Bi-LSTM for training. The structure diagram is shown in Fig. 3.

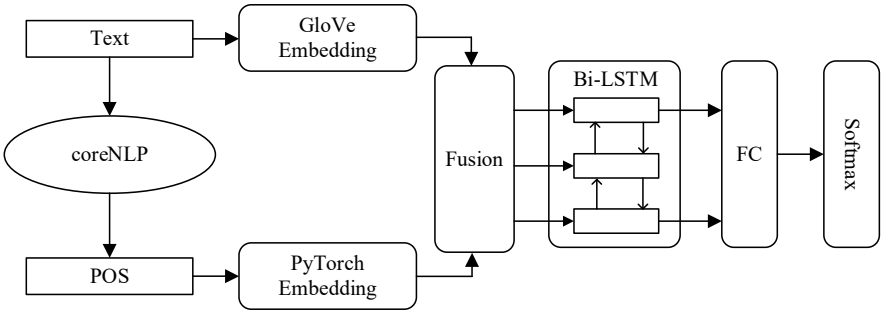

Fig. 3. Early fusion structure diagram based on part-of-speech features and semantic features

3.4.2 Late Fusion of POS Features and Semantic Features

Unlike early fusion, late fusion first uses a single feature to train the model separately, and then merges the results of multiple model training. The advantage of this method is that the results of the model can be selected flexibly, which improves the fault tolerance of the system; the amount of fusion information calculation is reduced and the real-time performance of the system is improved. There are two classic late fusion methods:

(1) Feature is not fused, multi-scale features are predicted separately, and then the prediction results are synthesized, such as Single Shot MultiBox Detector (SSD), Multi-scale CNN (MS-CNN);

(2) Feature performs pyramid fusion, and then performs prediction after fusion, such as Feature Pyramid Network (FPN), etc. [27].

The late fusion in this article uses the second method. We first input the part-of-speech features and semantic features into Bi-LSTM respectively, then stitch the results of the two models, and finally input them into the fully connected layer. The structure diagram is shown in Fig. 4.

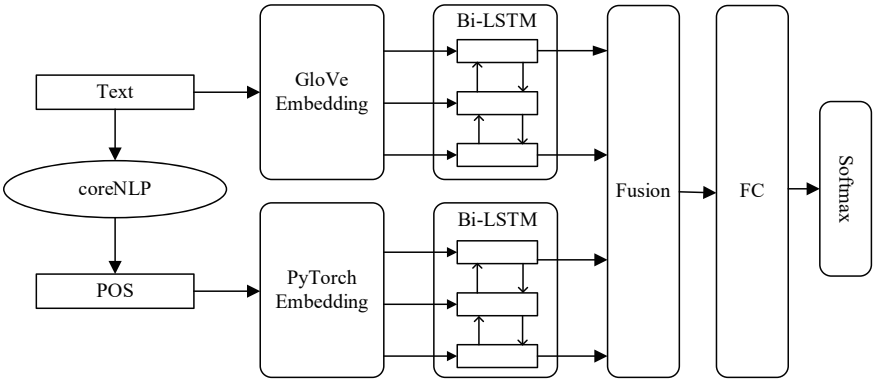

Fig. 4. Late fusion structure diagram based on part-of-speech features and semantic features

4 Experiment and Analysis

4.1 Experimental Data

In the training phase, we use the academic phrase library in the Academic Phrasebank website created by Dr. John Morley of the University of Manchester, which contains a total of 2865 academic formulaic language. Then we download 10 papers, preprocess the text, choose the sentences with the smallest PMI, and get a total of 4500 sentences. The data set consisting of positive samples and negative samples is divided into training set, validation set and test set, with a ratio of 8:1:1.

We download 50 papers in the computer field, extract text parts other than references, preprocess the text, delete formulas, special symbols, etc., to make it more structured. Then we segment the sentences and observe the characteristics of the sentences. The sentence length statistics are shown in Fig. 5. As can be seen from the figure, the sentence length is concentrated around 25 words, so the sentence with the number of words less than 3 and more than 40 is deleted, and finally, 10802 sentences are obtained as the data of the prediction stage.

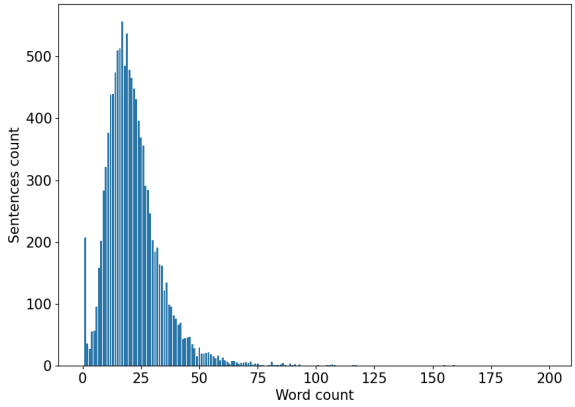

Fig. 5. Sentence length statistics graph

4.2 Experimental Setup and Result Analysis

4.2.1 Parameter Settings

For the trained academic formulaic language, the 300-dimensional word vector pre-trained by Glove is used as the semantic input feature. For the part-of-speech feature, the dimension of the word embedding vector generated by the embedding layer is set to 300 dimensions. We adopt mini-batch stochastic gradient descent, the batch size is set to 20. The learning rate is set to 0.001. The decay ratio is set to 0.9, and the optimization algorithm selects the Adam algorithm. All LSTM networks have 128 neurons in a single layer, so the double-layer LSTM is 256,

which is trained for 100 rounds.

4.2.2 Experimental Evaluation Index

This article uses Accuracy (ACC), Precision, Recall, and F1-Score as the evaluation indicators of the experiment, where ACC represents the probability that the prediction is correct among all the predicted samples; Precision represents the proportion of samples that are correctly predicted among the samples that are identified as positive samples; Recall represents the proportion of all positive samples that can be predicted correctly; F1-Score is an indicator that neutralizes Precision and Recall. The formulas of *ACC*, *Precision*, *Recall*, *F1-Score* are as follows:

$$ACC=\frac{TP+TN}{TP+FP+FN+TN}\times 100\% . \tag{6}$$

$$Precision=\frac{TP}{TP+FP}\times 100\% . \tag{7}$$

$$Recall=\frac{TP}{TP+FN}\times 100\% . \tag{8}$$

$$F1=\frac{2\times Precision\times Recall}{Precision+Recall}\times 100\% . \tag{9}$$

In the formula, TP (True Positives) indicates that the prediction is positive and in fact it is also a positive case; FP (False Positives) indicates that the prediction is a positive case but it is actually a negative case; FN (false Negatives) indicates that the prediction is negative but in fact it is a positive case; TN (True Negatives) indicates that the prediction is negative and is actually a negative case.

4.2.3 Analysis of Results

Ashok [19] only uses GloVe as the semantic feature input into the model when recognizing the formulaic language. For comparison, in this paper, the recognition of academic formulaic language is understood as a binary classification problem, the existing academic formulaic language is trained with Bi-LSTM, and the classification results are obtained through the softmax function. In order to verify the effect of different characteristics and different fusion methods on the recognition of academic formulaic language, experiments were carried out through the Bi-LSTM model using the combinations of [part-of-speech features], [semantic features], [early fusion of part-of-speech features and semantic features] and [late fusion of part-of-speech features and semantic features].

The results of the four different combinations are shown in Table 3, where TPR represents the probability of correct prediction of the positive class, and TNR represents the probability of correct prediction of the negative class. It can be seen from the table that the Bi-LSTM model based on semantic features and the Bi-LSTM model based on late fusion have the highest TPR, and the Bi-LSTM model based on early fusion and the Bi-LSTM model based on late fusion have the highest TNR. We use a confusion matrix to evaluate the generalization performance of the model. The confusion matrices of the four comparative experiments are shown in Fig. 6 to Fig. 9.

The Bi-LSTM model based on part-of-speech features has the lowest ACC, Precision, Recall, and F1-Score, indicating that part-of-speech features cannot fully represent the characteristics of academic formulaic language. The results of the early fusion and the semantic feature model are similar, but from the results, the F1-Score of the semantic feature model is slightly higher. Through a comprehensive analysis of other values, it can be concluded that: after the splicing of part-of-speech features and semantic features in the early fusion, part-of-speech features will have a certain impact on the recognition of semantic features, making the error rate higher than that of the semantic feature model. The four values of late fusion based on part-of-speech features and semantic features all reached the highest values, respectively 98.16%, 99.09%, 97.21%, and 98.14%, indicating that this combination method has a more obvious effect on the prediction of academic formulaic language. The advantage of late fusion compared with early fusion is that it can flexibly select the results of part-of-speech feature model and semantic feature model. If the semantic features are more important, the semantic feature model can be given higher weight

to make the results more accurate.

Table 3. Experimental results of four different combinations

	TPR	FNR	FPR	TNR	ACC	Precision	Recall	F1-Score
Bi-LSTM_semantics	**97.21**	2.79	1.56	98.44	97.83	98.42	**97.21**	97.81
Bi-LSTM_POS	94.08	5.92	2.89	97.11	95.60	97.02	94.08	95.53
Bi-LSTM_early	96.17	3.83	0.89	**99.11**	97.64	99.08	96.17	97.60
Bi-LSTM_late	**97.21**	2.79	0.89	**99.11**	**98.16**	**99.09**	**97.21**	**98.14**

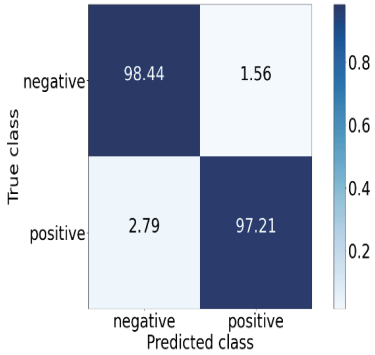

Fig. 6. Confusion matrix based on semantic feature model

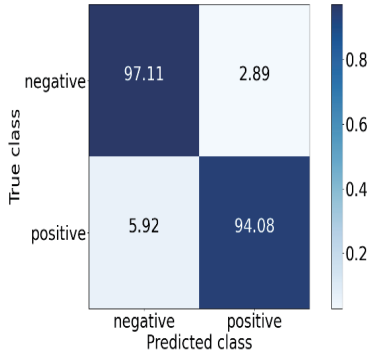

Fig. 7. Confusion matrix based on POS feature model

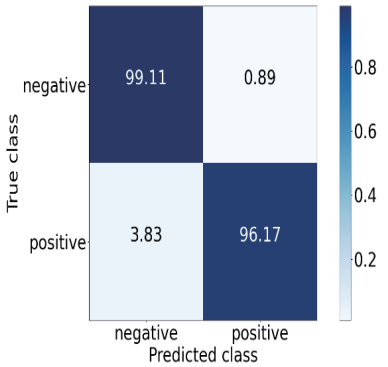

Fig. 8. Confusion matrix based on early fusion

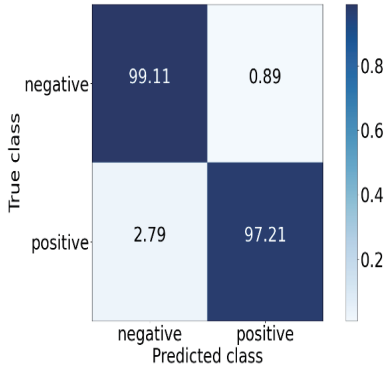

Fig. 9. Confusion matrix based on late fusion

Fig. 10 shows the change in accuracy of extracting different features from training data and test data into the model, where the solid line represents the accuracy of the training data, and the dotted line represents the accuracy of the test data. The observation shows that the accuracy increases gradually with the increase of the number of iterations, and finally tends to be stable. According to the results of the test data displayed in the dotted line, the accuracy of the late fusion method is the highest among the four combination methods. Fig. 11 shows the change of the loss function during training and testing, where the solid line represents the loss function in the training phase, and the dotted line represents the loss function in the testing phase. It can be seen from the figure that as the number of iterations increases, the loss function gradually tends to converge, and the loss function of the late fusion method is the smallest of the four combinations. It is known from the training curve that as the training progresses, the constructed model is being optimized towards the goal of the classification task.

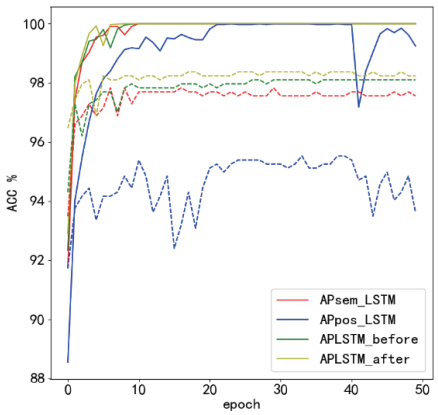

Fig. 10. Accuracy curve of training data and test data

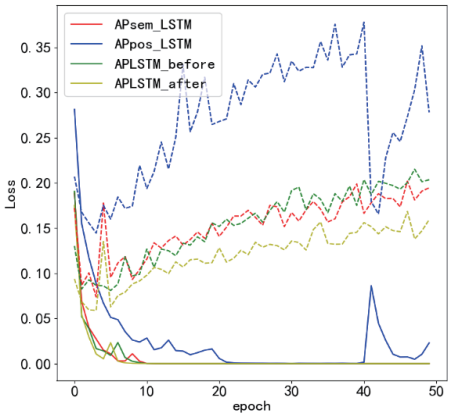

Fig. 11. Loss function curve of training data and test data

4.3 Prediction Stage

Predicting paper data refers to using a trained model to predict whether the sentences in the paper contain academic formulaic language, which saves a lot of manpower and material resources and does not require domain experts to read the entire paper. The model filters out sentences containing academic formulaic language, and then expands the corpus of formulaic language.

Since the classification results of four different features are compared in the training stage, the comparison experiment shows that the classification effect of the late fusion method is the best, so the prediction stage of this paper adopts the feature fusion method of late fusion. We download 50 papers in the computer field, extract the text parts other than references, and preprocess the text. Finally, a total of 10802 sentences are obtained. Part-of-speech features and semantic features are extracted from the sentence, then the features are fused through late fusion and input into the model to predict whether academic formulaic language is included. The output of the model is the sentence and the predicted probability value. The probability value represents the probability that a sentence contains academic formulaic language. Finally, the threshold is selected, and the sentences whose probability value is greater than the threshold are selected and stored in the academic formulaic language corpus.

In the process of threshold selection, count the number of samples in each probability range, calculate the Precision, Recall, and F1-Score respectively, and select the probability with the highest F1-score as the threshold. Generally speaking, it is to predict the number of accurate sentences as much as possible, and the number of missing sentences containing academic formulaic language as little as possible. The statistical results are shown in Fig. 12. It can be seen from the figure that the F1-Score with the probability of a sentence containing academic formulaic language greater than 0.5 is the highest, so 0.5 is selected as the threshold.

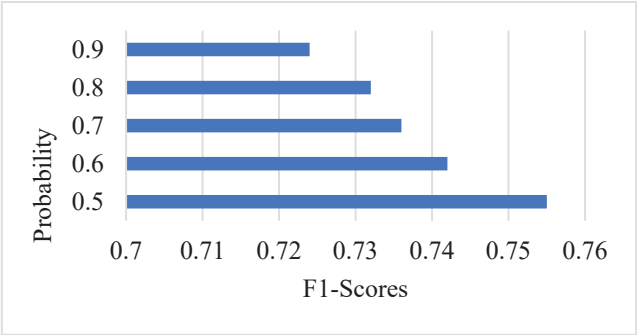

Fig. 12. F1-Score statistics corresponding to different probabilities

In addition, this paper also compares the prediction results of different features and different fusion methods on academic formulaic language. We input the part of speech features, semantic features, late fusion of part-of-

speech features and semantic features, and early fusion of part-of-speech features and semantic features of new sentences into the trained classifier model to predict whether the sentences contain academic formulaic language. When comparing the results, if the selected result range is small, the results of the semantic feature-based model, the early fusion model and the late fusion model will not be much different, but the model based on the part-of-speech feature will be different from them. Therefore, the late fusion model and the part-of-speech-based model are selected as examples, and the same part of the same article is selected to compare the prediction results of the two models. The results are shown in Table 4 and Table 5.

Table 4. Examples of results of model based on part-of-speech feature

Formulaic language	Bi-LSTM_POS
declarative sentence	In fact, they do not update at all, but rather stick to a fixed opinion.
are introduced in Section II.	The topics of NLP and AI, including deep learning, are introduced in Section II.
are used to	These solutions are used to build useful software.
declarative sentence	Currently, NLP is primarily a data-driven field using statistical and probabilistic computations along with machine learning.
is proposed to	A novel texture pattern representation method called Hierarchical Visual Codebook is proposed to encode the distinctive and robust texture primitives of iris images.

Table 5. Examples of results of model based on late feature

Formulaic language	Bi-LSTM_after
are introduced in Section II.	The topics of NLP and AI, including deep learning, are introduced in Section II.
such as that present in X,	Attention mechanisms, such as that present in X, allow the decoder to determine which portions of the encoding are most relevant at each output step.
In addition, can be developed for	In addition, a module of iris image classification can be developed for iris recognition systems for various applications inspired by the general framework.
are composed of	Neural Networks and Deep Learning Neural networks interconnected nodes, or neurons, each receiving some number of inputs and supplying an output.
is proposed to	A novel texture pattern representation method called Hierarchical Visual Codebook is proposed to encode the distinctive and robust texture primitives of iris images.

By comparing the sentences in the two tables, we find that the late fusion sentences have richer semantics and more types of sentence patterns. However, some of the results of the model based on part-of-speech features are declarative sentences, which have no semantic and pragmatic functions.

4.4 Experimental Verification

The main content of the academic formulaic language prediction model based on multi-feature fusion is to judge whether the sentences contain academic formulaic language. The purpose is to narrow the search scope of the corpus, so as to identify academic formulaic language more efficiently. It can screen the sentences with high probability containing academic formulaic language from large-scale texts, and improve the efficiency of academic formulaic language recognition task. In order to verify the effectiveness of the prediction model, we first input 10000 sentences into the academic formulaic language recognition model, and the processing time is 334.93s; Secondly, 10000 sentences are input into the prediction model, and then the results are input into the recognition model. The processing time is 194.60s. It can be seen that the efficiency of the recognition task can be improved.

5 Conclusion

This paper proposes a predictive model of academic formulaic language based on multi-feature fusion, which uses a late fusion method based on part-of-speech features and semantic features to predict academic formulaic language. The experimental results show that, because the model can select part-of-speech features and semantic features according to the degree of importance, it is not only more flexible than the early fusion method, but also the prediction results are more accurate. Nevertheless, when using this model to process real paper data, the

prediction accuracy rate is much lower than the experimental results. The reason may be that the sentence structure of the real paper data is more complex, the semantic types are more, and it is more difficult to be accurately recognized. In addition, Hyland studied papers published in four disciplines (electrical engineering, microbiology, business, and applied linguistics) and found that there are great differences in the formulaic language used by different disciplines [28]. Therefore, in future research, we should extract more features that can represent academic formulaic language, so that the model can predict and recognize academic formulaic language with higher accuracy on real essays. Secondly, since this article only uses papers in the computer field as data, it will involve multiple fields in the future and be applied to guide academic writing, machine translation and other practical work in different fields.

6 Acknowledgement

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), 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), Jilin City Science and Technology Innovation Development Project “Study on the Emotion Classification of Jilin Tourism Online Review Text” (No. 20200104108).

References

[1] G. Li, An overview of foreign research on formulaic language identification, Journal of Central South University 18(5) (2012) 221-225.
[2] L.-X. Yu, Corpus-based formulaic language research at home and abroad-review and summary, Overseas English (9) (2019) 221-222.
[3] X.-R. Hu, C.-Q. Yao, Y.-F. Gao, Phrase Recognition Method Combining Multiple Strategies, Information Science 37(6) (2019) 49-54.
[4] X. Zhang, H.-Y. Sun, D.-X. Xin, C.P. Li, H. Chen, Survey on automatic term extraction research, Journal of Software 31(7) (2020) 2062-2094.
[5] C. Strobl, E. Ailhaud, K. Benetos, A. Devitt, O. Kruse, A. Proske, C. Rapp, Digital support for academic writing: A review of technologies and pedagogies, Computers & Education 131(2019) 33-48.
[6] J. Liu, F. Tang, Y. Liu, An Extraction Method for Chinese Terminology Based on Statistical Technology, China Terminology 16(5)(2014) 10-14.
[7] P. Pecina, Lexical association measures: Collocation extraction, the 4th publication in the series Studies in Computational and Theoretical Linguistics, Institute of Formal and Applied Linguistics, 2009.
[8] G.C. Pastor, J.-P. Colson (Eds.), Computational Phraseology, John Benjamins, 2020 (pp. 83-110).
[9] K. Hyland, Academic clusters: Text patterning in published and postgraduate writing, International Journal of Applied Linguistics 18(1)(2008) 41-62.
[10]Y.-H. Chen, P. Baker, Lexical bundles in L1 and L2 academic writing, Language Learning & Technology 14(2)(2010) 30-49.
[11]R. Simpson, N.-C. Ellis, An academic formulas list: New methods in phraseology research, Applied Linguistics 31(4) (2010) 487-512.
[12]M.-B. O’donnell, The adjusted frequency list: A method to produce cluster-sensitive frequency lists, ICAME Journal 35(2011) 135-169.
[13]X. Zhang, Research on Automatic Recognition of Noun Phrase Structure Based on Rules, Journal of Jilin Teachers Institute of Engineering and Technology 29(7)(2013) 70-72.
[14]K. Iwatsuki, F. Boudin, A. Aizawa, Extraction and evaluation of formulaic expressions used in scholarly papers, Expert Systems with Applications 187(2022) 115840.
[15]D. Liu, The most frequently-used multi-word constructions in academic written English: A multi-corpus study, English for Specific Purposes 31(1)(2012) 25-35.
[16]P. Durrant, J. Mathews-Aydınlı, A function-first approach to identifying formulaic language in academic writing, English for Specific Purposes 30(1)(2011) 58-72.
[17]M.-A. Abbas, S. Hammad, G.-J. Hwang, S. Khan, S.M.M. Gilani, An assistive environment for EAL academic writing using formulaic sequences classification, Interactive Learning Environments (2020) 1-15.
[18]W. Gharbieh, V. Bhavsar, P. Cook, Deep learning models for multiword expression identification, in: Proc. of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017), 2017.
[19]A. Ashok, R. Elmasri, G. Natarajan, Comparing Different Word Embeddings for Multiword Expression Identification, in: Proc. International Conference on Applications of Natural Language to Information Systems, 2019.

[20]N. Klyueva, A. Doucet, M. Straka, Neural networks for multi-word expression detection, in: Proc. of the 13th workshop on multiword expressions (MWE 2017), 2017.
[21]A. Fazly, P. Cook, S. Stevenson, Unsupervised type and token identification of idiomatic expressions, Computational Linguistics 35(1)(2009) 61-103.
[22]J. Pennington, R. Socher, C.-D. Manning, Glove: Global vectors for word representation, in: Proc. of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014.
[23]J.-L. Elman, Finding structure in time, Cognitive Science 14(2)(1990) 179-211.
[24]S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9(8)(1997) 1735-1780.
[25]N. Xiong, P. Svensson, Multi-sensor management for information fusion: issues and approaches, Information Fusion 3(2)(2002) 163-186.
[26]B. Khaleghi, A. Khamis, F.O. Karray, S.N. Razavi, Multisensor data fusion: A review of the state-of-the-art, Information Fusion 14(1)(2013) 28-44.
[27]F. Xiao, Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy, Information Fusion 46(2019) 23-32.
[28]K. Hyland, As can be seen: Lexical bundles and disciplinary variation, English for Specific Purposes 27(1)(2008) 4-21.