Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 7622392, 10 pages
https://doi.org/10.1155/2022/7622392

Research Article

A Data-Driven Model for Automated Chinese Word Segmentation and POS Tagging

Qing Xu [1] **and Zhiyou Wang** [2]

[1]*Changsha University of Science and Technology, Changsha, Hunan 410000, China*
[2]*School of Electronic Communication and Electrical Engineering, Changsha University, Changsha, Hunan 410000, China*

Correspondence should be addressed to Qing Xu; 1531040158@xzyz.edu.cn

Received 11 August 2022; Revised 25 August 2022; Accepted 5 September 2022; Published 16 September 2022

Academic Editor: D. Plewczynski

Chinese natural language processing tasks often require the solution of Chinese word segmentation and POS tagging problems. Traditional Chinese word segmentation and POS tagging methods mainly use simple matching algorithms based on lexicons and rules. The simple matching or statistical analysis requires manual word segmentation followed by POS tagging, which leads to the inability to meet the practical requirements for label prediction accuracy. With the continuous development of deep learning technology, data-driven machine learning models provide new opportunities for automated Chinese word segmentation and POS tagging. Therefore, a data-driven automated Chinese word segmentation and POS tagging model is proposed in order to address the above problems. Firstly, the main idea and overall framework of the proposed automated model are outlined, and the tagging strategy and neural network language model used are described. Secondly, two main optimisations are made on the input side of the model: (1) the use of word2Vec for the representation of text features, thus representing the text as a distributed word vector; and (2) the use of an improved AlexNet for efficient encoding of long-range word, and the addition of an attention mechanism to the model. Finally, on the output side, an additional auxiliary loss function was designed to optimise the Chinese text based on its frequency. The experimental results show that the proposed model can significantly improve the accuracy and operational efficiency of Chinese word segmentation and POS tagging compared with other existing models, thus verifying its effectiveness and advancement.

1. Introduction

Over the past decades, natural language processing has received much attention in the field of machine learning, and its applications are widespread, such as data mining, system management, and opinion monitoring. With the rapid development of the contemporary information society and the increasing amount of textual information generated by the Internet, natural language processing techniques are becoming increasingly important. These vast amounts of data exist in different forms on the Internet and managing them using automated natural language processing methods is a daunting and commercially valuable task. Natural language is the language that humans need to use in their daily lives, studies, and work [1–5]. Natural language is a fundamental method of communication between people. Unlike artificially designed machine languages, natural languages are highly abstract, complex, and versatile. Natural language processing is the core technology that enables the exchange of information between humans and machines and is the backbone of artificial intelligence.

Currently, natural language processing techniques can be broadly divided into five levels [6–9]: phonological analysis, lexical analysis, syntactic analysis, semantic analysis, and pragmatic analysis. The purpose of lexical analysis is to determine the position, composition, and properties of words in a sentence according to syntactic rules. For Chinese natural language processing, lexical analysis mainly consists of two tasks: word segmentation and part-of-speech (POS) tagging [10–12]. Unlike phonetic writing, which is based on

the Latin alphabet, Chinese is an ancient meaning-phonetic writing. Since there are no natural spaces between words in a sentence, word segmentation is a unique and important part of natural language processing in Chinese. POS tagging requires that each word be given a lexical label. POS is the most basic grammatical feature of words, indicating their composition and role in a sentence [13, 14]. Word segmentation and POS tagging are the most fundamental techniques in natural language processing, apart from speech analysis. Word segmentation and POS tagging have a crucial impact on upper-level tasks such as syntactic analysis, text classification, and machine translation.

On the other hand, deep learning, a branch of machine learning, has once again seen a surge in research as the amount of data has exploded and computing power has increased [15–18]. Compared with traditional statistical-based machine learning, data-driven deep learning no longer requires researchers to spend too much time and effort on feature engineering. Data-driven deep learning can automatically learn and extract features from large amounts of data; thus, unlocking the potential value of large amounts of data [19–22]. In view of the great success of deep learning in the field of computer vision, many researchers have started to try to use deep learning techniques to solve natural language processing problems and have gradually accumulated a lot of research experiences and application results. As the underlying core technology of Chinese natural language processing, the study of Chinese word segmentation and POS tagging has great theoretical and practical significance. However, deep learning has not achieved significant advantages over traditional statistical analysis methods in the area of Chinese word segmentation and POS tagging, suggesting that there is still room for further research and optimisation of the problem.

Therefore, the purpose of this research is to apply the data-driven deep learning technology to Chinese word segmentation and part-of-speech tagging tasks, so as to achieve the automation of these two tasks at the same time with high precision.

The rest of the paper is organised as follows: related works are presented in Section 2. In Section 3, the proposed data-driven automation-based model was studied in detail, while Section 4 provides the optimisation of the input layer. The optimisation of the output layer is presented in Section 5, while Section 6 provides the experimental results and analysis. Finally, the paper is concluded in Section 7.

2. Related Work

Research on Chinese word segmentation began in the 1990s. The earliest Chinese word segmentation methods were lexicon- and rule-based methods, such as forward maximal matching, reverse maximal matching, and least word cut. Although lexicon- and rule-based methods were easier to implement, the accuracy of word segmentation was low. Hong et al. [23] proposed to use the Google News corpus to handle lexicon-based Chinese word segmentation tasks. Chang et al. [24] introduced conditional random fields into Chinese word segmentation, which has become the mainstream method for Chinese word segmentation. Most of the word segmentation methods based on word-related feature learning use semiconditional random fields, and thus need to assign the same label to multiple consecutive words in each step, resulting in long training time.

POS tagging refers to assigning each word in a sentence a category called part-of-speech. POS is the most basic grammatical property of a vocabulary. Accurate POS tagging will bring great advantages to upper-level tasks such as syntactic analysis. The earliest approaches to POS tagging were rule-based methods. Although rule-based approaches have a relatively simple theory, they require a lot of time to manually write various complex rules. Wang et al. [25] proposed an error-driven machine learning method that automatically learns lexical transformation rules; thus, reducing the workload of writing rules manually to a certain extent. With the development of statistical machine learning, POS tagging was modelled as a sequence tagging problem. The algorithms that can be used for the sequence tagging problem can basically be used to solve the POS tagging problem, such as support vector machines, hidden Markov models, maximum entropy models, and conditional random fields. Tursun et al. [26] used hidden Markov models for English POS tagging and achieved 95% tagging accuracy.

With the rise of deep learning, deep neural network-based POS tagging methods have also received much research. Liu et al. [27] first proposed a neural network-based Chinese word segmentation method in 2019, verifying the feasibility of deep learning on the Chinese word segmentation task. Su et al. [28] proposed a recurrent neural network with gate structure and used it to extract n-gram features in the Chinese word segmentation task. Deep learning-based Chinese word segmentation methods have been the mainstream approach in the field of Chinese word segmentation research in recent years, and many successful experiences have been achieved. However, there is still room for improvement in the accuracy and training efficiency of deep learning-based word segmentation methods compared to traditional word segmentation methods.

AlexNet is the first deep convolutional neural network model for image classification [29–34] and is a historically important model in the field of convolutional neural networks. Therefore, this paper uses the AlexNet model to simultaneously perform Chinese word segmentation and POS tagging in one complete step. In order to improve the model accuracy and operational efficiency, the input and output of the model are optimised in this paper.

The main innovations and contributions of this paper include the following:

(1) The AlexNet network was improved in the input to the model to reduce the number of model parameters, resulting in a faster training and learning speed. In addition, an attention mechanism was added to the AlexNet network to efficiently learn the contextual semantics of the target words. Word weights are adjusted during the training process based on the relevance of the input to the word vector to the final predicted outcome.

(2) Additional auxiliary loss functions were designed to optimise the output of the model based on the Chinese text frequencies.

3. Data-Driven Automation-Based Model

In the field of Chinese natural language processing, word segmentation and POS tagging are regarded as two separate processing steps. In other words, for a Chinese sentence, a word segmentation model is used to segment the words, and then a POS tagging model is used to annotate the lexical properties of the segmented words. This approach is also known as the pipeline model. In this model, word segmentation and POS tagging are two steps, one after the other, and the POS tagging depends on the outcome of the segmentation while the segmentation is not affected by the POS tagging in any way. This pipeline model has two obvious drawbacks: the problem of error transmission and word segmentation cannot sense POS information. The former refers to the fact that errors in word segmentation are passed on to POS tagging. The latter refers to the fact that the POS information is not directly considered in the segmented model itself. To address these two issues, this paper decides to use a joint model that is entirely word-based. This paper argued that word segmentation has a large impact on POS tagging and that they should not be considered as two separate processing steps. Therefore, this paper attempts to reduce the error rate of POS tagging by using only one model to deal with both Chinese word segmentation and POS tagging at one time.

3.1. Overall Model Architecture Design. The overall architecture of the joint model proposed in this paper is shown in Figure 1.

Firstly, the input layer needs to represent each word as a vector. This is because the model in this paper is entirely word-based and therefore word vectors must be used as the basic input to the network. In this paper, text word features are extracted using word2Vec so that the text is represented as a distributed word vector. Considering the complex combinatorial relationships of Chinese words, an AlexNet network model is also used for further feature extraction of the word vectors after the input of the distributed word vectors.

The hidden layer refers to the intermediate layer in a broad sense, rather than the hidden layer in a general feed-forward neural network. In order to make full use of the syntactic and semantic information contained in the original corpus, this paper introduces a neural network language model as a secondary task to be trained jointly with the main task. The output layer predicts a label for each word in the sentence. In order to explicitly model the dependencies between tags, the classical conditional random field is used in the output layer for tag prediction. At the same time, an auxiliary loss function is added to the output side of the model for predicting the word frequencies at the corresponding positions; thus, improving the training effect of the whole model.

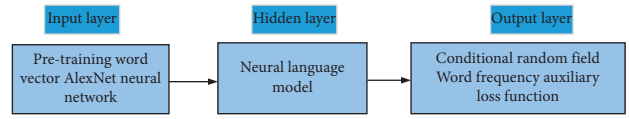

Figure 1: Joint model architecture.

Table 1: Examples of Chinese word segmentation tag sets.

	Tag	Description
Quaternary tag set	*B*	Beginning
	M	Middle
	E	Ending
	S	One-character words
Ternary tag set	*B*	Beginning
	I	Middle or end
	0	One-character words
Binary tag set	*B*	Beginning
	NB	Not the beginning

3.2. Labeling Strategy. The Chinese word segmentation tags indicate the relative position of a character in a word, as shown in Table 1, and mainly include binary tag sets, ternary tag sets, and quaternary tag sets. The greater the number of tags, the greater the number of positions that can be represented, but at the same time the complexity of the model increases and the efficiency of both training and prediction decreases. Conversely, the smaller the number of tags, the more efficient the model, but the less accurate it is. Considering the accuracy and efficiency of the model, this paper uses a quaternary tags set (*B*, *I*, *E*, and *S*) to represent the boundaries of a word [35]. *B* means the current character is the first character of the word, *M* means the current character is in the middle of the word, *E* means the current character is the last character of the word, and *S* means the current character is a single word.

3.3. Neural Network Language Models. Current supervised sequence tagging methods based on manually annotated corpora tend to make use of the tagging information alone, without making use of the information in the original corpus itself. The size of the manually annotated corpus is very limited due to cost and financial constraints; in contrast, the size of the unannotated raw text is getting larger and larger. Therefore, how to make full use of the unannotated original text becomes the key issue to be considered in this paper. Word vector-based methods such as word2vec are a shallow approach. The models are not able to capture the high-level information of natural language in the pretraining process of word vectors by using word2vec techniques alone. Neural network language models can effectively solve this problem, as shown in Figure 2.

Pretrained word vectors based on large-scale raw text are an effective way to extract information from the original paper, and the resulting word vectors have some ability to characterise the semantics. However, as the lowest-level input to the model, the pretrained word vector itself is only a shallow feature representation and lacks richer high-level information. Therefore, this paper proposes the use of neural

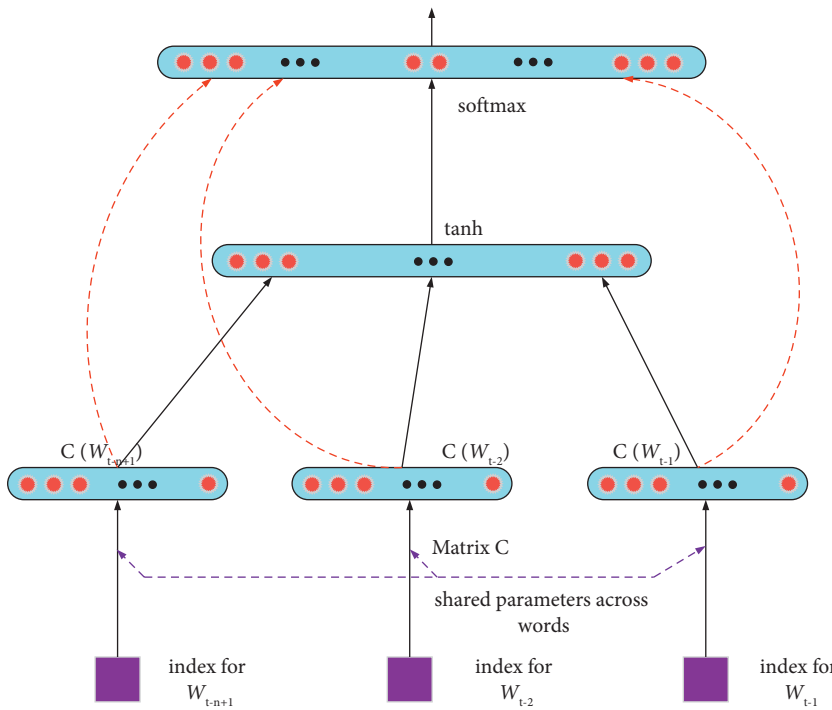

Figure 2: Architecture of neural network language model.

network language models as an alternative method of extracting information from raw text.

$$P_f(c_1, \cdots, c_n) = \prod_{i=1}^{N} P_f\left(c_i \mid c_1, \cdots, c_{i-1}\right), \tag{1}$$

where c_i indicates the character to be predicted at the current moment and $c_1, \cdots, c_{i-1}$ indicates all the characters in the text.

$$P_f\left(c_i \mid c_1, \cdots, c_{i-1}\right) = \frac{\exp\left(w_{c_i}^T f_{i-1}\right)}{\sum_{\widehat{c_j}} \exp\left(w_{\widehat{c_j}}^T f_{i-1}\right)}, \tag{2}$$

where w_{C_i} represents the weight vector corresponding toc_i and f_{i-1} represents the output of the neuron. Thus, the training objective function of neural network can be obtained.

$$F = -\frac{1}{N} \sum_i \log P_f(c_i). \tag{3}$$

3.4. Conditions Follow the Airport. In the input layer, for each word of a sentence, there is a corresponding word vector. After a hidden layer consisting mainly of a neural network language model, we are given a corresponding output vector. The output vector models the contextual information of the sentence adequately. The next step is to predict the final output labels based on the output vector. There are obvious dependencies between the output labels, so it is often necessary to explicitly model the label dependencies at the output layer. Conditional random fields (CRFs) [36–38] are suitable for exactly this need. A CRF is a model of conditional probability distribution. By manually designing suitable feature templates, conditional random fields can obtain a strong feature fitting capability, resulting in a large number of feature functions. Conceptually, CRF is a probabilistic undirected graph model.

For the input sentence$x = (c_1, \cdots, c_n)$, the corresponding sequence of output labels is $y(z) = (y_1, \cdots, y_n)$. Given z, a CRF probability model can express the conditional probability of y.

$$p(y \mid z; W, b) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, z)}{\sum_{y' \in Y(z)} \prod_{i=1}^{n} \psi_i(y'_{i-1}, y'_i, z)}, \tag{4}$$

where $\psi_i(y_{i-1}, y_i, z)$ denotes the potential energy function in the CRF. W and b denote the weight vector and bias vector corresponding to the label(y', y), respectively. In the training phase, maximum likelihood estimation is used with the objective of maximising the conditional likelihood.

$$\mathcal{J}_{CRF} = -\sum_i \log p\left(y_i \mid z_i\right). \tag{5}$$

In the test or decoding phase, the objective is to search for the globally optimal sequence of labels y^*, that is, to require the maximum conditional probability.

$$y^* = \underset{y \in Y(z)}{\operatorname{argmax}}\, p(y \mid z; W, b). \tag{6}$$

4. Optimisation of the Input Layer

4.1. Word Vector Representation of Text. The distributed representation of the word vector word2Vec [39] mainly consists of CBOW and skip-gram, where the structure of

CBOW is very similar to that of a neural network language model, as shown in Figure 3. CBOW uses a strategy where all words share a hidden layer to achieve efficient model training. The input to CBOW is contextual and requires vector summation. Skip-gram predicts context based on the central word. Skip-gram can be represented as maximising the log conditional probability.

The CBOW model consists of three layers: input, projection, and output. For the word sequence $(\mathscr{W}_{t-2}, \mathscr{W}_{t-1}, \mathscr{W}_t, \mathscr{W}_{t+1}, \mathscr{W}_{t+2})$, w_t is the current word and the rest of the words are their contexts. The projection layer is used for the accumulation of input layers, thus obtaining X_w. Random negative sampling is performed in the output layer to achieve the prediction. Assuming that the given context w *is* a positive sample and the other words are negative samples, any word v can be represented as follows:

$$L^w(v) = \begin{cases} 1, v = w \\ 0, v \neq w \end{cases}, \tag{7}$$

where $L^w(v)$ denotes the label of word v. 1 represents a positive sample, and 0 represents a negative sample. For a given positive sample $(\text{Context}(w)w)$, the final goal is to maximize G.

$$G = \log_2 \prod_{w \in C} g(w), g(w) = \prod_u p(u \mid \text{Context}(w)). \tag{8}$$

In general, if the probability of a negative sample decreases, the probability of a positive sample increases. Given the context, CBOW will derive the target words. In this paper, the CBOW model will be used to train the word vectors. Word2vec word vector training parameters are shown in Table 2.

In addition, as the traditional word vector would ignore the order of words and their relationships in the text, the training time will be too long. Therefore, this paper modifies the text representation model on the basis of the traditional word representation, as shown in Figure 4. Topic categories between different texts are trained using LDA estimation. The topic categories are combined with the top-ranked words to generate a hybrid representation model. Related studies have shown that this hybrid representation can improve prediction accuracy and reduce training time. In addition, this hybrid representation is suitable for unbalanced data classification.

4.2. Improved AlexNet Based on Attention Mechanism. First, word2Vec was used to word represent the text and train word vectors; thus, representing natural language in the form of distributed vectors. Then, a one-dimensional convolution was used in the convolutional layer, which aimed to reduce the number of text encoding features. Next, a modified AlexNet was used to efficiently encode long-range words. Finally, the text feature representations are fed into the attention layer, which selects features that are highly relevant to the final classification result as the final output.

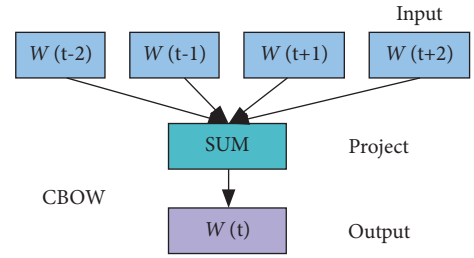

Figure 3: CBOW neural network model.

Table 2: Word2vec word vector training parameters.

Parameter name	Meaning of parameter	Parameter value
Alpha	Initial learning rate	0.01
Size	Vector dimension	64
Mincount	Minimum word frequency	5
Window	Context window size	15
Negative	Number of negative samples	20

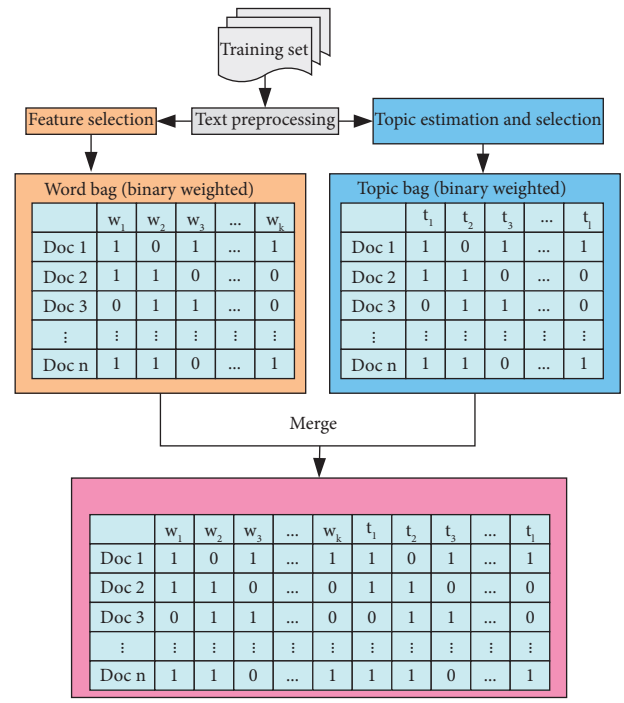

Figure 4: Hybrid representation.

AlexNet is a derivative model of CNN. However, the training time for AlexNet is long; therefore, this paper makes further improvements to the AlexNet network. The improved AlexNet replaces the location of the fully connected layer with a convolutional layer and removes the LRN layer. Figure 5 shows the structure of the improved AlexNet, which is divided into 8 layers: layers 1 to 5 are convolutional layers, while layers 6 to 8 are fully connected layers. The activation function is a ReLU function, which acts as a nonsaturating nonlinear function to speed up the convergence of the model. Dropout layers are added after layer 6 to prevent

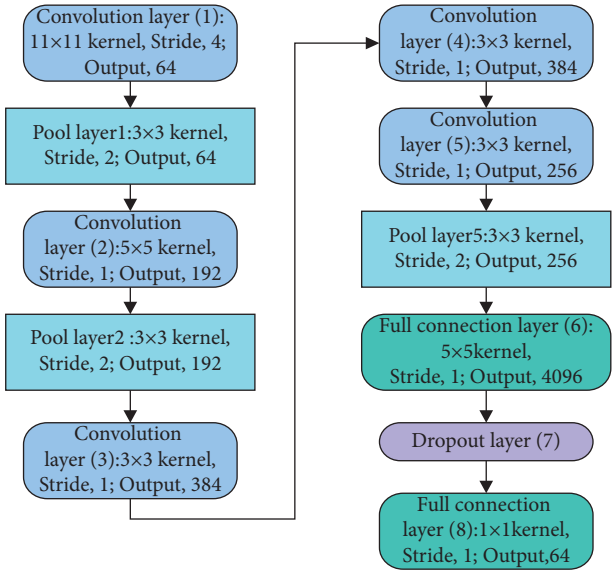

Figure 5: Structure of the improved AlexNet.

overfitting problems. The final layer has no activation function. The improved AlexNet inherits many of the features of AlexNet, such as easier parallelisation operations.

The role of the convolutional layer is to extract the semantics of the text. It is generally possible to preserve the convolutional layers and pretrained parameters during the extraction process from shallow semantics to advanced semantics. In this paper, the first five conv layers of AlexNet are preserved and some FC layers are removed. The parameters of the improved AlexNet are shown in Table 3. The improved AlexNet can quickly extract the underlying semantic features from the original text and reduce the number of dimensions.

Attentional mechanisms were first used to model the attentional behaviour of the human brain [40, 41]. In the improved AlexNet, the conv layer narrows down the input features used in prediction. However, for all input words, the correlation between each word and the final prediction is not the same. The use of a one-dimensional convolutional layer reduces the number of features needed to encode text.

The improved AlexNet in this paper receives the features generated in the conv layer stage and generates them by extracting the last hidden layer. The improved AlexNet is able to access contextual information and treats the obtained information as two different text representations. The text features are fed into the improved AlexNet model, which generates a feature representation of the sequence. This feature representation is then fed into the attention layer, which selects the features that are highly relevant to the final classification result. Using such processing, the attention mechanism can efficiently learn the contextual information surrounding the target word, thereby significantly improving prediction accuracy and reducing the number of weights required for prediction.

Bahdanau attention model is used in attention mechanism. Based on the hidden vector of word vectorsh_j and the weights α_{ij}, the vector of contextual words for the i-th target word c_i can be calculated.

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j. \tag{9}$$

The formula for calculating the weights is shown as follows:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T} \exp\left(e_{ik}\right)}. \tag{10}$$

4.3. Loss Functions. The improved AlexNet model was trained using a minimized cross-entropy loss function so as to fit a linear softmax classifier.

$$\text{Loss}(W) = -\sum_{i \in Jt} \sum_{k} y_{i,k} \ln\left(p\left(y_{i,k} \mid v_i; W\right)\right), \tag{11}$$

where W denotes *the* network model weights and v_i ($i \in \{1,\ldots, \mathrm{n}\}$) denotes the attribute of the i-th text.

5. Optimisation of the Output Layer

Looking at the word segmentation results, it was found that there were a very high number of occurrences of common words in the training corpus. The model was sufficiently trained on the frequently used words, so the tagging results were more accurate. However, due to the relatively small number of occurrences of rare words, the model is not sufficiently trained on rare words, resulting in poor tagging results, which can be regarded as a kind of "Hard Examples". This small number of hard examples can lead to a decrease in the overall performance of the model. If we can improve the accuracy of the tagging of the rare characters, the overall performance of the model will also be improved. It was found that the frequency of Chinese characters approximately conforms to Ziff's law. This empirical law has been validated in many fields and is also known broadly as the two-eight law.

A small number of high-frequency words appear very often, and a large number of low-frequency words appear very rarely. The low-frequency characters are therefore the "Hard Examples" of the task. Considering the characteristics of this character frequency distribution and not wanting the model to be too complex, this paper proposes a simple approach to solve this problem. An auxiliary loss function is added to the output side of the model to predict the word frequencies at the corresponding positions. This loss function is intended to aid the training of the whole model and is expressed as a squared difference loss function.

$$\mathcal{J}_{\text{char\_freq}} = \sum_{i} \left(f_i - \widehat{f}_i\right)^2, \tag{12}$$

where f_i indicates the predicted word frequency of the *i-th character* in the sentence and $\widehat{f}$ indicates the true word frequency of the *i-th* character.

Table 3: Parameters of the improved AlexNet.

Layer	Convolution kernel size	Step length	Number of cores
Convolutional layer 1	$11 \times 11 \times 3$	4	96
Pooling layer 1	$1 \times 3 \times 3 \times 1$	2	—
Convolutional layer 2	$5 \times 5 \times 48$	1	256
Pooling layer 2	$1 \times 3 \times 3 \times 1$	2	—
Convolution layer 3	$3 \times 3 \times 384$	1	256
Convolutional layer 4	$3 \times 64 \times 384$	1	384
Convolutional layer 5	$3 \times 3 \times 256$	1	384
Pooling layer 5	$1 \times 3 \times 3 \times 1$	2	—
Fully connected layer 6	32×256	—	256
ReLU6	—	—	384
Dropout layer	—	—	—
Fully connected layer 7	32×256	—	256
ReLU7	—	—	256
Dropout layer	—	—	—
Fully connected layer 8	256×2	—	2
Prob	Softmax	—	—
Output	—	—	—

By adding the auxiliary loss function described above, the model is explicitly guided to predict word frequencies. The auxiliary loss function gives the model the ability to distinguish between high- and low-frequency words, which improves the accuracy of the model for low-frequency words. The experiments show that the auxiliary loss function does have a positive effect on the tagging results.

6. Experimental Results and Analysis

6.1. Experimental Parameter Settings. In this paper, a data-driven automated Chinese word segmentation and POS tagging model is implemented using the Python programming language and Google's TensorFlow deep learning framework. The key parameters of the automated model are shown in Table 4. The model is trained using gradient descent-based backpropagation, where the batch size is set to 32 and the maximum number of training sessions is 30. The number of convolutional layers used for word feature extraction is set to 3.

The larger the dimension of the word vector, the better the model training results. However, when the dimension of the word vector is large enough, the increase in the dimension of the word vector is very small. However, the larger the word vector dimension is, the slower the model can be trained and tested. Therefore, we choose to set the word vector dimension to 64 to find a good compromise between model effectiveness and speed.

Dropout is an extremely common technique in the field of deep learning and is very important in mitigating model overfitting; Dropout is to some extent similar to L2 regularisation; the core idea of Dropout is to drop a number of neurons at random with a certain probability during model training, so that the weights of the neurons can represent various aspects of the input data, rather than concentrating on certain elements. This method introduces random noise, which can be used to create a new model. This method of introducing random noise can increase the sparsity of the network and has the effect of alleviating model overfitting. Experiments in this paper found that dropping neurons at random with a 20% probability worked best.

Table 4: Key parameters of the automation model.

Parameter	Numerical value
Number of AlexNet hidden layer nodes	200
AlexNet network depth	2
Number of AlexNet convolutional layers (pooling layers)	3
Dropout	0.2
Training optimiser	AdaGrad
Initial learning rate	0.1
Batch size	32
Word vector dimension	64

6.2. Experimental Data and Evaluation Indicators. Three different publicly available datasets were used to test the proposed automated model, as shown in Table 5. Ninety percent of each dataset was randomly selected as the training set, while the remaining 10% was used as the test set.

This paper evaluates the performance of the automation model using the micro-$F1$ metric (micro-$F1$) and the macro-$F1$ metric (macro-$F1$). Higher values of micro-$F1$ and macro-$F1$ indicate stronger performance.

$$\text{Micro\_F1} = \frac{2 \times p_t \times r_t}{p_t + r_t}, \tag{13}$$

where p_t indicates the total accuracy and r_t indicates the total recall.

$$\text{Macro}_{\text{F1}} = \frac{2 \times p_m \times r_m}{p_m + r_m}, \tag{14}$$

where p_m represents the mean of accuracy and r_m represents the mean of recall.

Table 5: Presentation of the four datasets.

Dataset	Data segmentation	Number of words (K)	Number of sentences (K)
MSR	Training set	494	18
	Test set	8.0	348
CTB7	Training set	78	31
	Test set	245	10
CTB5	Training set	2131	78
	Test set	107	4

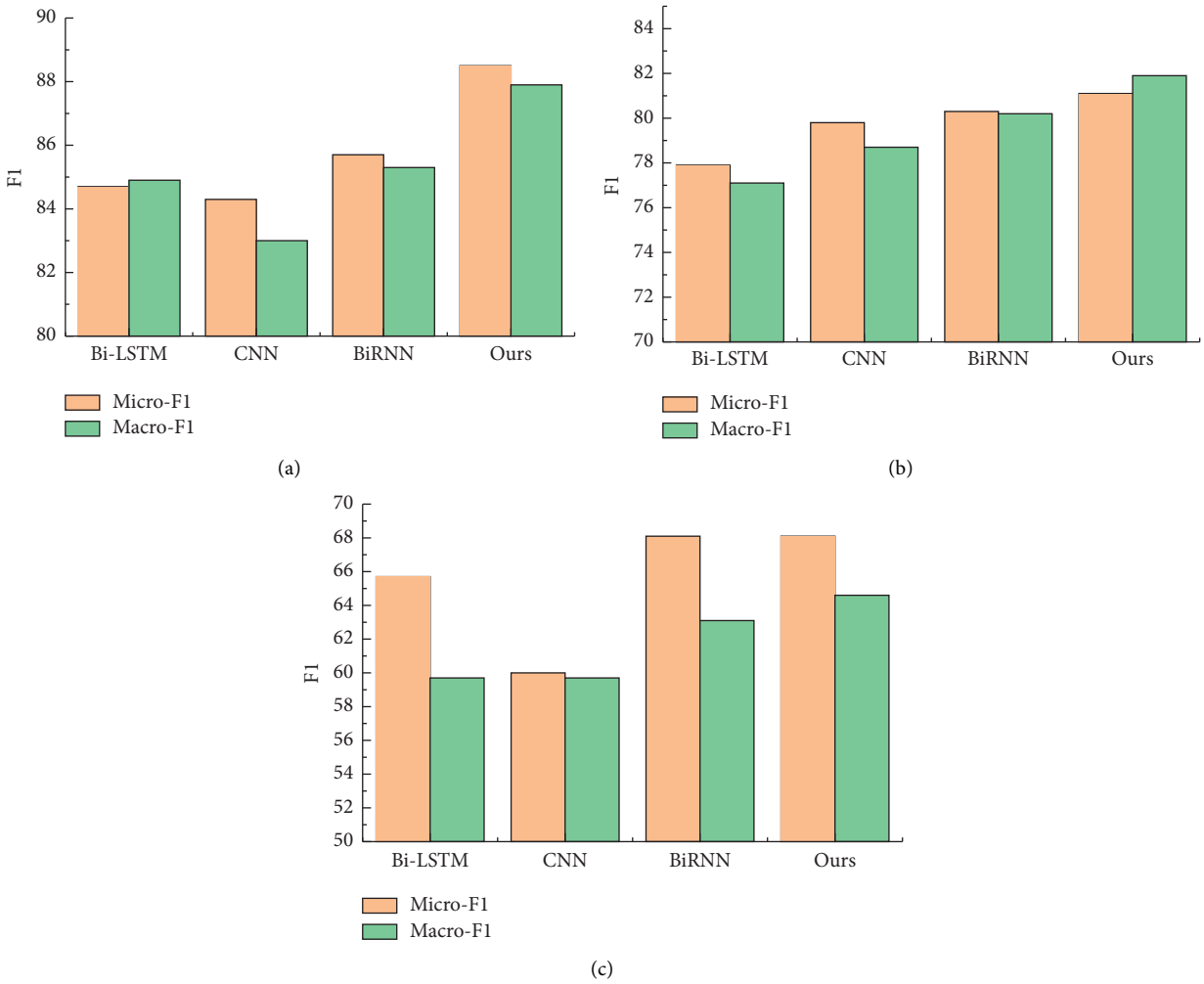

Figure 6: Accuracy comparison results: (a) MSR data set, (b) CTB7 dataset, and (c) CTB5 dataset.

6.3. Performance Comparison. The proposed automated model was compared with Bi-LSTM [42], CNN [43], and BiRNN [44] on the above three publicly available datasets, and the experimental results are shown in Figure 6. It can be seen that the performance of the method in this paper on two datasets (MSR and CTB7) is significantly better than the other methods. On the CTB5 dataset, the BiRNN method is closer to the method in this paper. Overall, this method achieves better text classification performance. This is mainly due to the fact that the proposed method adds an attention mechanism to the AlexNet-2 architecture, which adjusts the weights according to the relevance of each input to the final prediction result.

A comparison of the average time required for the training of the four models is shown in Figure 7. It can be seen that the proposed automated model is slightly faster than the Bi-LSTM model. Although both models use an attention mechanism, the improved AlexNet framework has fewer parameters and a lighter structure. The two models, CNN and BiRNN, take more time to tune the parameters, resulting in longer running times.

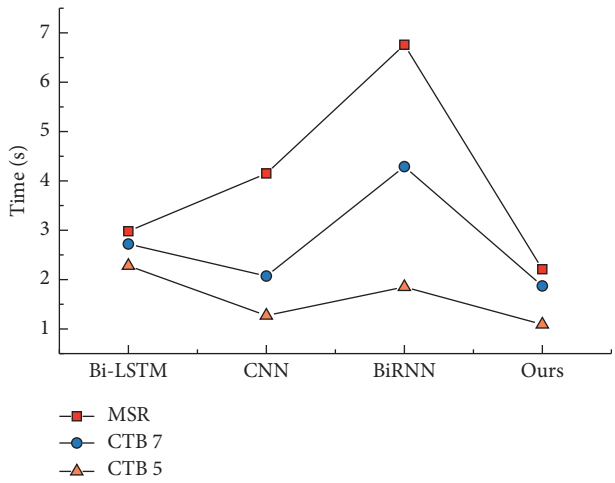

Figure 7: Comparison of the average time spent in training.

7. Conclusion

In order to implement automated Chinese word segmentation and POS tagging, this paper proposes a data-driven automation model. The proposed model consists of three parts: an input layer, a hidden layer, and an output layer. The input layer uses word2Vec to extract textual word features, and a modified AlexNet network model is used to further extract features from the word vectors. The hidden layer introduces a neural network language model as a secondary task for joint training with the main task. The classical conditional random field was used in the output layer for label prediction. At the same time, an auxiliary loss function is added to the output side of the model for predicting the word frequency at the corresponding position; thus, enhancing the training effect of the whole model. The experimental results show that the proposed model has a certain improvement in both micro-$F1$ metrics and macro-$F1$ metrics compared with other models and also has a high operational efficiency.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

The study was supported by Research on the Application and Practice of Medical Esp Teaching for English Majors under the Situation of New College Entrance Examination Reform (Project No.: XJK20CGD048) and the Research on the Path of Ideological and Political Reform of English Major Courses (Project No.: 20c0216).

References

[1] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011.

[2] A. Chopra, A. Prashar, and C. Sain, "Natural language processing," *International Journal of Technology Enhancements and Emerging Engineering Research*, vol. 1, no. 4, pp. 131–134, 2013.

[3] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016.

[4] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4291–4308, 2021.

[5] N. Y. Habash, "Introduction to Arabic natural language processing," *Synthesis Lectures on Human Language Technologies*, vol. 3, no. 1, pp. 1–187, 2010.

[6] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021.

[7] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis Lectures on Human Language Technologies*, vol. 10, no. 1, pp. 1–309, 2017.

[8] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X Huang, "Pre-trained models for natural language processing: a survey," *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, 2020.

[9] C. Friedman, L. Shagina, and Y. Lussier, "Automated encoding of clinical documents based on natural language processing," *Journal of the American Medical Informatics Association*, vol. 11, no. 5, pp. 392–402, 2004.

[10] C. Friedman, L. Shagina, Y. Lussier, and G Hripcsak, "Automated encoding of clinical documents based on natural language processing," *Journal of the American Medical Informatics Association*, vol. 11, no. 5, pp. 392–402, 2004.

[11] M. D. Yandell and W. H. Majoros, "Genomics and natural language processing," *Nature Reviews Genetics*, vol. 3, no. 8, pp. 601–610, 2002.

[12] A. Farzindar and D. Inkpen, "natural language processing for social media," *Synthesis Lectures on Human Language Technologies*, vol. 8, no. 2, pp. 1–166, 2015.

[13] M. Lapata and F. Keller, "Web-based models for natural language processing," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 2, no. 1, pp. 3–16, 2005.

[14] S. R. Joseph, H. Hlomani, and K. Letsholo, "Natural language processing: a review," *International Journal of Research in Engineering and Applied Sciences*, vol. 6, no. 3, pp. 207–210, 2016.

[15] L. Zhou, C. Zhang, F. Liu, Z. Qiu, and Y He, "Application of deep learning in food: a review," *Comprehensive Reviews in Food Science and Food Safety*, vol. 18, no. 6, pp. 1793–1811, 2019.

[16] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: a survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.

[17] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C. W Lin, "Deep learning on image denoising: an overview," *Neural Networks*, vol. 131, pp. 251–275, 2020.

[18] A. Esteva, A. Robicquet, B. Ramsundar et al., "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, 2019.

[19] S. Dargan, M. Kumar, M. R. Ayyagari, and G Kumar, "A survey of deep learning and its applications: a new paradigm to machine learning," *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1071–1092, 2020.

[20] D. A. Bashar, "Survey on evolving deep learning neural network architectures," vol. 2019, no. 2, pp. 73–82, 2019.

[21] J. Chen and X. Ran, "Deep learning with edge computing: a review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[22] Y. Ma, D. Yu, and T. Wu, "PaddlePaddle: an open-source deep learning platform from industrial practice," *Frontiers of Data and Domputing*, vol. 1, no. 1, pp. 105–115, 2019.

[23] C. M. Hong, C. M. Chen, and C. Y. Chiu, "Automatic extraction of new words based on Google News corpora for supporting lexicon-based Chinese word segmentation systems," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3641–3651, 2009.

[24] C. H. Chang, H. M. Chuang, C. Y. Huang, Y. S. Su, and S. Y Li, "Enhancing POI search on maps via online address extraction and associated information segmentation," *Applied Intelligence*, vol. 44, no. 3, pp. 539–556, 2016.

[25] Y. Wang, S. Liu, N. Afzal et al., "A comparison of word embeddings for the biomedical natural language processing," *Journal of Biomedical Informatics*, vol. 87, pp. 12–20, 2018.

[26] E. Tursun, D. Ganguly, T. Osman et al., "A semisupervised tag-transition-based markovian model for Uyghur morphology analysis," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 16, no. 2, pp. 1–23, 2016.

[27] J. Liu, F. Wu, C. Wu, Y. Huang, and X. Xie, "Neural Chinese word segmentation with dictionary," *Neurocomputing*, vol. 338, pp. 46–54, 2019.

[28] B. Su and S. Lu, "Accurate recognition of words in scenes without character segmentation using recurrent neural network," *Pattern Recognition*, vol. 63, pp. 397–405, 2017.

[29] S. Sadhana and R. Mallika, "An intelligent technique for detection of diabetic retinopathy using improved alexnet model based convoluitonal neural network," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 1, pp. 1–12, 2020.

[30] B. Cong, H. Ling, and P. Xiang, "Optimization of deep convolutional neural network for large scale image retrieval," *Neurocomputing*, vol. 303, pp. 60–67, 2018.

[31] J. Liu, Z. Xie, C. Zhang, and G. Shi, "A novel method for Mandarin speech synthesis by inserting prosodic structure prediction into Tacotron2," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 10, pp. 2809–2823, 2021.

[32] F. Leite, A. Akcamete, B. Akinci, G. Atasoy, and S. Kiziltas, "Analysis of modeling effort and impact of different levels of detail in building information models," *Automation in Construction*, vol. 20, no. 5, pp. 601–609, 2011.

[33] J. Abualdenien and A. Borrmann, "Levels of detail, development, definition, and information need: a critical literature review," *Journal of Information Technology in Construction*, vol. 27, pp. 363–392, 2022.

[34] H. Ismail Fawaz, B. Lucas, G. Forestier et al., "InceptionTime: finding AlexNet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[35] Z. Luo, J. Zhu, and Z. Li, "Research the Method of Joint Segmentation and POS Tagging for Tibetan Using BiGRU-CRF," in *Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 1–6, Sanya, China, December 2020.

[36] F. Peng and A. Mccallum, "Information extraction from research papers using conditional random fields," *Information Processing & Management*, vol. 42, no. 4, pp. 963–979, 2006.

[37] F. Mahmood and N. J. Durr, "Deep learning and conditional random fields-based depth estimation and topographical reconstruction from conventional endoscopy," *Medical Image Analysis*, vol. 48, pp. 230–243, 2018.

[38] M. T. Abd and M. Mohd, "A comparative study of word representation methods with conditional random fields and maximum entropy Markov for bio-named entity recognition," *Malaysian Journal of Computer Science*, vol. 31, no. 5, pp. 15–30, 2018.

[39] O. Kolesnikova and A. Gelbukh, "A study of lexical function detection with word2vec and supervised machine learning," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 2, pp. 1993–2001, 2020.

[40] X. Xu, "Research on neural network machine translation model based on entity tagging improvement," *Mathematical Problems in Engineering*, vol. 2022, Article ID 8407437, 8 pages, 2022.

[41] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2Vec model analysis for semantic similarities in English words," *Procedia Computer Science*, vol. 157, pp. 160–167, 2019.

[42] H. Ye, B. Cao, Z. Peng, T. Chen, Y. Wen, and J Liu, "Web services classification based on wide & Bi-lstm model," *IEEE Access*, vol. 7, pp. 43697–43706, 2019.

[43] T. Kattenborn, J. Leitloff, F. Schiefer, and S Hinz, "Review on convolutional neural networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, 2021.

[44] J. Wu, H. Woo, and Y. Tamura, "Pedestrian trajectory prediction using BiRNN encoder–decoder framework," *Advanced Robotics*, vol. 33, no. 18, pp. 956–969, 2019.