Hindawi
Scientific Programming
Volume 2022, Article ID 8187680, 11 pages
https://doi.org/10.1155/2022/8187680

Research Article

Deep Neural Networks Algorithm for Vietnamese Word Segmentation

Kexiao Zheng[1] **and Wenkui Zheng**[2]

[1]*School of Language and Literature, Guilin University, Guilin 541006, China*
[2]*College of Computer and Information Engineering, Henan University, Kaifeng 475004, Henan, China*

Correspondence should be addressed to Wenkui Zheng; henuzwk@henu.edu.cn

Received 29 December 2021; Revised 26 January 2022; Accepted 31 January 2022; Published 28 February 2022

Academic Editor: Baiyuan Ding

Traditional Vietnamese word segmentation methods do not perform well in the face of Vietnamese ambiguity, in response to the enormous challenge posed by the scarcity of the Vietnamese corpus to language processing. We first investigated the most advanced deep neural network method. According to the ambiguity problem of Vietnamese word segmentation, we then proposed a Vietnamese word segmentation processing technology based on an improved long short-term memory neural network (LSTM), which is made up of an LSTM encoding and a CNN feature extraction portion. The previous important information is kept in the memory unit; the word segmentation processing task is refined into a classification problem and a sequence labeling problem, which can gain the useful features of the word segmentation character and word level automatically. The limitation of the local context window size is avoided, and the word segmentation processing task is refined into a classification problem and a sequence labeling problem. Finally, validated by a homemade Vietnamese news website crawler dataset, the experimental results show that, compared with the single LSTM, single CNN methods, and traditional methods, the performance improvement of our proposed method is more obvious. In the Vietnamese word separation task, the accuracy reaches 96.6%, the recall reaches 95.2%, and the F1 value reaches 96.3%, which is significantly better than the traditional methods CNN and LSTM.

1. Introduction

Under the current dual promotion of economic globalization and artificial intelligence. In the subject of machine translation, language processing has become an important technique, and language processing technology is based on language word segmentation. At present, in the field of linguistic information processing, there has been a lot of studies on word segmentation. The research findings are divided into three types: dictionary-based word segmentation methods, statistics-based word segmentation methods, and understanding-based word segmentation methods. The dictionary-based word segmentation method matches the character string to be studied with the entries of a machine dictionary that has been artificially created according to a strategy. If it is successfully matched with the string in the character dictionary, following that, word segmentation is carried out. The statistical method mainly performs statistical analysis on the words and phrases in the corpora and calculates the information about their mutual occurrence. The closeness of the characters' combination relationship is reflected in the mutual information. When the gap between the characters is greater than a specific threshold, this character combination can be deemed a word. Finally, by defining the mutual information of two characters, then calculating the probability of the two characters appearing next to each other, through algorithm design, the understanding-based word segmentation method allows the computer to imitate a human's understanding of the text, in order to reach the appearance of words being recognized. It is challenging to organize varied linguistic information into a form that can be directly read by a machine due to the generality and complexity of language knowledge.

Currently, in the realm of linguistic analysis, using information retrieval techniques, various studies have been completed for common languages such as English and Chinese [1, 2], hand-crafted rules [3], or neural mechanisms [4–6]. However, there has been minimal research into Vietnamese word segmentation. Most research on Vietnamese is limited to detecting the meaning of word segmentation using traditional methods [7] or using connection matching to extract context [8]. As far as we know, in the study of Vietnamese word segmentation, research at home and abroad has just begun. Until now, there are no specific shared resources available for academic research. All language resources need to be built from scratch. As the basis of Vietnamese natural language processing, Vietnamese word segmentation requires the collection of Vietnamese corpus resources and processing them as required, which is a prerequisite for Vietnamese word segmentation. At present, the most widely used Vietnamese word segmentation tool is VnTokenizer [9], which was developed by the University of Hanoi in 2008 based on the maximum matching and N-Gram model [10]. Vietnamese word segmentation currently has two major problems: combination ambiguity and cross ambiguity. Although some researchers have used maximum entropy, SVM, and CRF [11] methods to segment Vietnamese words and have achieved certain success, they are all in the experimental stage. The accuracy is not stable enough. Based on previous work, combining Vietnamese word-formation features and language features, we propose a model based on long short-term memory neural network (LSTM), which is determined by input, output, and forgetting gates of how to use previous information to model and update the memory of previous information. Experiments show that the method presented in our research is capable of effectively resolving the ambiguity issue. This paper's contribution can be summed up as follows.

(i) Firstly, we introduced the relevant research work in the direction of language processing and then proposed the study of Vietnamese word segmentation in response to the scarcity of the Vietnamese corpus.

(ii) Although traditional methods are not effective in processing Vietnamese word segmentation ambiguity models, we have studied methods based on deep neural networks for Vietnamese word segmentation.

(iii) Although Vietnamese word segmentation is a special case, our model is an improvement of the LSTM method, which makes our method easy to generalize and apply to other sequence labeling tasks.

(iv) A Vietnamese word segmentation model based on an enhanced LSTM framework is presented, which is composed of an LSTM encoding part and a CNN feature extraction part.

(v) The findings of the experiments reveal that, when compared to the old method and the single LSTM and the single CNN method, our method shows a greater improvement in performance.

The remainder of this paper is arranged in the following manner. Section 2 discusses language processing-related work. Section 3 introduces the Vietnamese language features and ambiguity model and then describes in detail the relevant principles and implementation process of the improved LSTM Vietnamese word segmentation processing method. Section 4 reports the experimental dataset, experimental settings, evaluation indicators, and analysis of experimental results. Finally, Section 5 summarizes our research and reveals some further research works.

2. Related Work

To detect language segmentation, most studies use annotated corpora to train traditional classifiers by investigating different types of features [12, 13]. However, this technique necessitates a significant amount of feature engineering. Another choice is to learn discriminative features by using deep neural network algorithms. The specific process is shown in Figure 1. First, input the vector, then use the filter to extract the character features and automatically learn, and then output the prediction results through the pooling layer and the fully connected layer. For instance, Shi et al. [14] presented modeling multilevel nonlinear feature representations by stacking distinct CNN feature maps; Kato et al. [15] used recurrent autoencoders to spoken Japanese conversations based on smartphones systematic Japanese word segmentation classification of discourse; Goo et al. [16] proposed a slotted gate, which concentrates on understanding the link between word segmentation and slot attention vectors in order to achieve superior semantic framework results via overall optimization.

To infer the meaning of word segmentation through joint context, the majority of contemporary research makes use of IR technologies [17]. Ji et al. proposed an optimization method that extracts the most comparable word segmentation problem from a specified common context and responds by matching the most likely word segmentation as a morpheme. This technology is usually used to build open domain chat interface machine translation (for example, to provide small chat services) or to answer common questions in a given domain. In terms of e-commerce, to select the most appropriate response from the existing datasets, Cui et al. [4] use the word segmentation autonomous learning system to support small chats and comments on the research and development system. Yan et al. [6] proposed a generic approach to building a task-based dialogue system for online shopping. To obtain the PI requested by the customer, the system uses the DSSM model to match the question with the basic segmentation PI [18]. However, many researchers do not permit online booking by customers; in many practical uses, external data resources are challenging to manage. The ultimate goal of these researchers is to build a cross-language retrieval system, as shown in Figure 2.

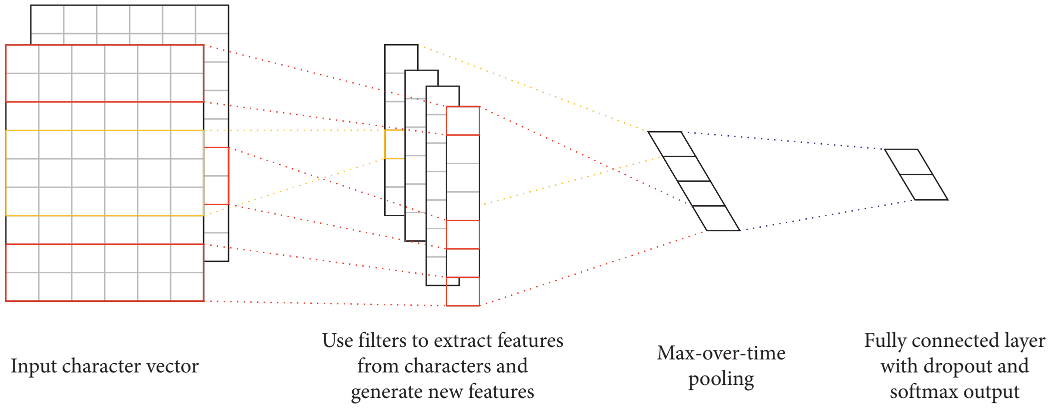
Input character vector
Use filters to extract features from characters and generate new features
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 1: Use deep neural methods to learn character features.

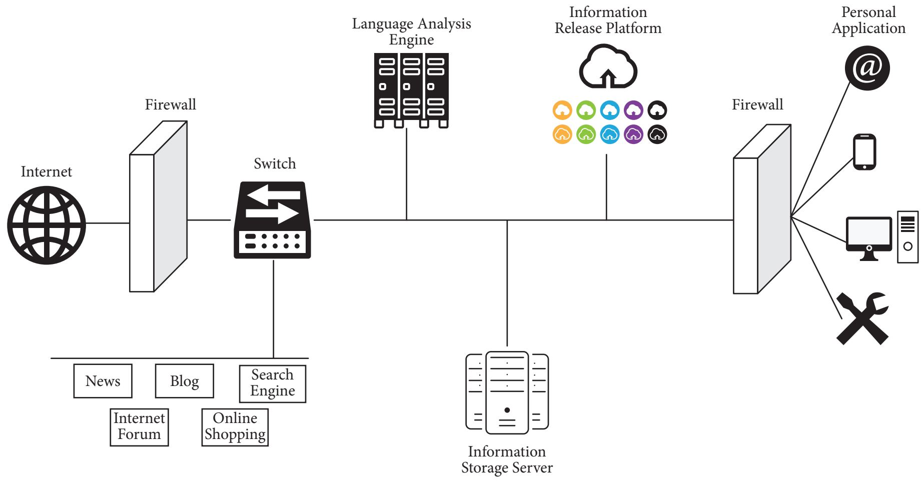

Figure 2: Cross-language retrieval system.

There has been very little research on Vietnamese word segmentation. The MaxEnt classifier is used by Ngo to allocate applications, and in order to detect sentence meanings, use the connected dictionary matching rule. [7]. This matching system is extremely expensive, necessitates domain expertise, and cannot handle Vietnamese ambiguity. In this work, due to simple problems being easier to analyze, they always focus on small problems to assist users in completing mobile phone interaction tasks. Tran et al. [8] investigated the recognition of named entities in spoken Vietnamese text. This research provides a simple machine learning model that incorporates a vast variety of handcrafted characteristics. However, designing these functions manually is extremely difficult; it also necessitates a high level of expertise in the realm of technology and the Vietnamese language.

There have been some research results in Vietnamese word segmentation methods [19–21]. However, there are still various challenges in understanding natural language, especially in the case of ambiguity. There are two main types of ambiguity in Vietnamese, which are combined ambiguity and cross ambiguity. Different participles will produce different ambiguities in Vietnamese expressions. In this paper, we concentrate on researching and comprehending the ambiguity in Vietnamese word segmentation to help language processing technology to better understand Vietnamese. As mentioned earlier, this language presents some challenges. To this end, we presented a solution using long and short-term memory networks. The solution uses input, output, and forget gates to determine how to use previous information to explicitly model and renew the memory of the previous feature. If a beneficial character from the input sequence is detected by the LSTM unit at a preliminary phase, it can easily change this feature to be carried long distances to capture potentially useful long-distance information. When new features are generated in the word segmentation traversal, they will be compared again, and then the corpus will be updated again to achieve a full understanding of the purpose of the Vietnamese sentence.

3. Method

3.1. Vietnamese Features and Ambiguity. The tones of Vietnamese are very similar to Chinese Pinyin. Each syllable is composed of initials, vowels, and tones, but the difference is that the initials of Vietnamese include the flat, sharp, profound, question, down, and accent. Like Chinese, although it lacks morphological changes, every syllable has meaning. In addition, its composition is the Latin alphabet, phonetic characters, and punctuation marks. Morphemes, as the word-building units of Vietnamese, can be divided into five categories: monosyllable words, compound words, accented alliterative words, coupled words, and derivative works. The construction of Vietnamese phrases also plays a vital role in Vietnamese participles. Transforming the Vietnamese phrase structure tree into a dependency structure tree is the current standard processing approach. The labeling system of the Vietnamese dependency structure tree library is shown in Figure 3, which annotates the dependencies and types of dependencies between words in a sentence [22, 23].

The annotation of sentences in the Vietnamese phrase structure tree library is shown in Figure 4. It only identifies the phrase hierarchy and phrase type of each sentence and does not indicate the central subnode of each phrase. The most common way to determine the central subnode of a phrase is to use the central subnode filter table. The dependency tree structure is a supplement to the phrase structure tree. The advantage of this is that the scale of the target tree bank can be increased, and the ability of the dependency analyzer can be improved without changing the learning strategy of the syntactic analysis model. In other words, it is a way of learning syntactic knowledge using multiple treebanks. It has a good experimental effect in dealing with Vietnamese treebank conversion and Vietnamese dependency treebank expansion and solves the problem of Vietnamese dependency syntax analysis well.

The development of the central subnode filtering table is an important part of the whole work. Table 1 is a part of the central subnode filtering table, and each row contains three items (phrase type, search direction, and priority), where phrase type is the phrase symbol for nonterminal nodes; search direction is the direction to search for the central subnode within nonterminal nodes. When the value is L, the search begins on the left side of the phrase and works its way to the right, and when the value is R, the search starts from the right side of the phrase to the left; priority is to determine the priority order of each token subnode as the central node in the phrase. For example, based on an entry in the filter table <VP, L, VP; V; A; AP; N; NP; S;. ∗>, the central subnode of the VP phrase can be determined as follows: observe each VPs from left to right to find a subnode, and the subnode with the symbol V found first is the central subnode of VP; if no VP node is found, observe each subnode of VP from left to right again, and the subnode with the symbol V found first is the central subnode of VP; and so on, if there is no subnode in VP with the symbols VP, V, A, AP, N, NP, S,. ∗, then the leftmost subnode is the central subnode by default.

There are two main types of Vietnamese ambiguity: combined ambiguity and cross ambiguity [24–26]. In combined ambiguity, some words are combined into sentences and have different meanings from word morphemes, such as "Bàn là một dụng cụ để ăn (The table is a tool for eating)." The morpheme "Bàn" means "table," and "là" means "is," but the combination of these two morphemes "Bàn là" means "iron." This ambiguity is called combined ambiguity. Cross ambiguity means that both the current morpheme and its preceding and following morphemes can form words, for example: "Tốc độ truyền thông tin không tốt lắm (transmission speed is not very good)," where "truyền thông" means "media," and "thông tin" means "information." This kind of ambiguity is cross ambiguity. These two kinds of ambiguous information often occur, which brings huge challenges to Vietnamese language processing.

3.2. Vietnamese Character Feature Extraction.

The selection of Vietnamese character features has a great impact on the result of word segmentation [27]. Combined with the morpheme characteristics of Vietnamese, we adopted a method based on a Markov random field method for character feature extraction. According to Tseng's research [10], in this paper, two basic features are selected, i.e., character N-gram feature and character repetition information feature, as shown in the following equation:

$$\begin{aligned}&\text{Character } N-\text{gram feature}\begin{cases} W_k\,(k=[-2,-1,0,1,2]) \\ W_kW_{k+1}\,(k=[-2,-1,0,1]) \end{cases},\\ &\text{Character repetition information feature}\begin{cases} W_kW_{k+2}\,(k=[-1,0]) \\ W_kW_{k-2}\,(k=[0,1]) \end{cases},\end{aligned} \tag{1}$$

where W represents the Vietnamese morpheme; W_0 represents the current morpheme, and k represents the position relative to the current morpheme. For example, in "Tôi thích nằm trên ghế và xem TV," if W_0 means the current Vietnamese morpheme "ghế," then W_{-1} means "trên"; W_{-2} means "nằm"; W_1 means "sofa." Repeat (W_0W_1) means that the current morpheme and the next morpheme are the same.

Aiming at the unregistered words that are prone to errors such as numbers, letters, and punctuation in Vietnamese, this paper defines Vietnamese morphemes into ten categories based on language characteristics: Sin, Pre, Suf, Pun, Dig, Let, Spe, Tim, Dat, and Oth [28]; related specific definitions are shown in Table 2.

3.3. Neural Model for Vietnamese Word Segmentation.

Vietnamese word segmentation is generally considered to be based on character sequence labeling. We use the Begin,

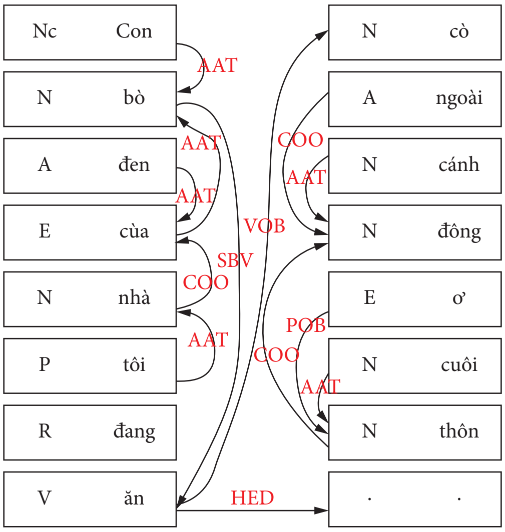

FIGURE 3: Example of Vietnamese dependency structure tree.

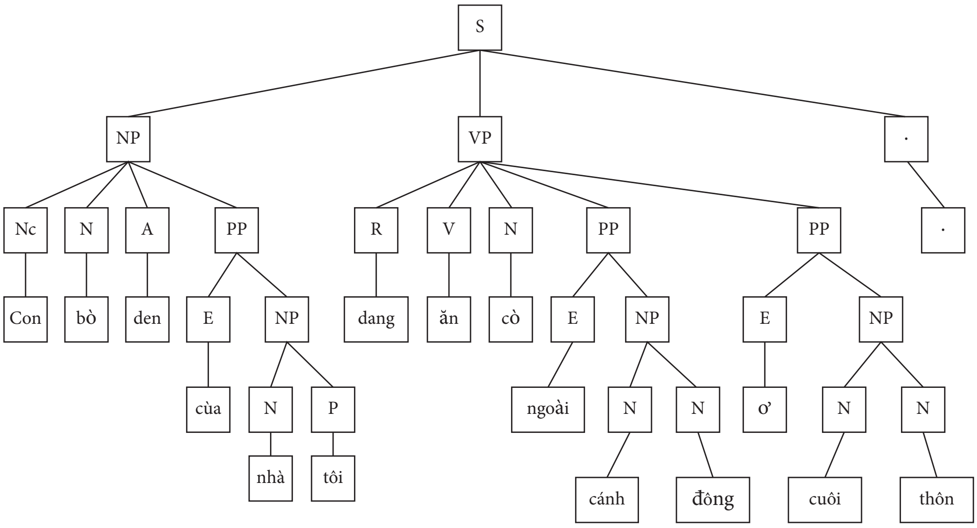

FIGURE 4: Example of Vietnamese phrase structure tree.

TABLE 1: Classification of Vietnamese characters.

Phrase type	Search direction	Priority
S	L	S; VP; AP; NP . ∗
SBAR	L	SBAR; S;VP; AP; NP;. ∗
SQ	L	SQ; VP; AP; NP;. ∗
NP	L	NP; Nc; Nu; Np; N;P. ∗
VP	L	VP; V;A; AP; N;NP; S . ∗
AP	L	AP; A;N; S . ∗
PP	L	PP; E;VP; SBAR; AP; QP . ∗
RP	R	RP; R;T; NP . ∗
XP	L	XP; X . ∗
MDP	L	MDP; T;I; A;P; R;X . ∗
UCP	L	. ∗
WHADV	L	R . ∗
WHVP	L	V . ∗
QP	L	QP; M . ∗

TABLE 2: Classification of Vietnamese characters.

Feature representation	Feature meaning	Example
Sin	Separate words	thi, ah
Pre	Start	c? m,xin
Suf	End	Sinh
Pun	Punctuation	,.!
Dig	Number	1,2,3
Let	Letters and combinations	A, a
Spe	Special identifier	@, %
Tim	Time	Minutes and seconds
Dat	Date	Date
Oth	Other	I II et al.

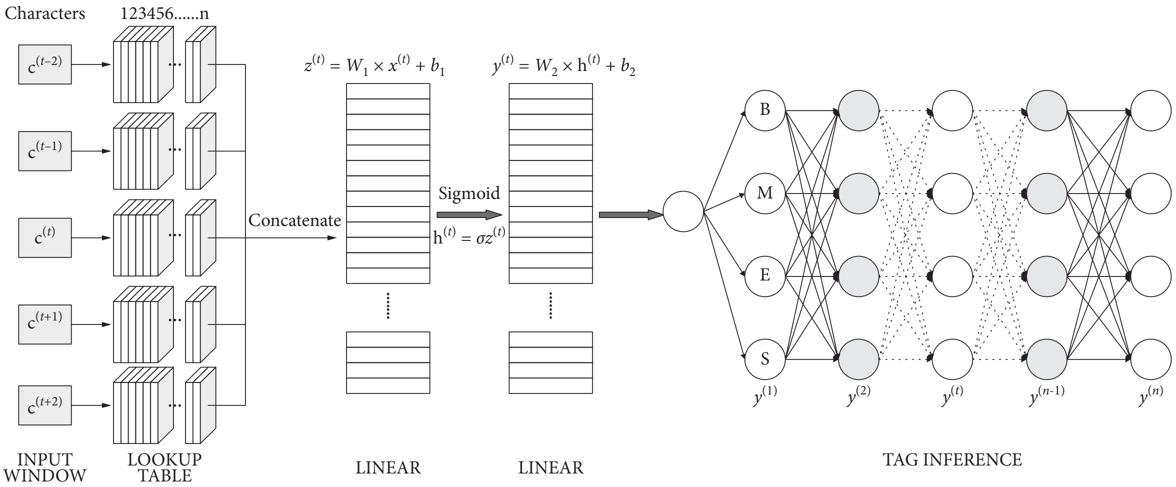

Figure 5: Vietnamese word segmentation neural model.

Middle, End, and Single of the multicharacter word segmentation to represent the Vietnamese characters and take the first letter to mark the sequence as {B, M, E, S}, and each Vietnamese character is marked as one of them, where the following neural model builds the foundation.

The marking method we usually use is the local window method. From the perspective of adjacent characters, it is assumed that the marked character has a close relationship with the weight of the neighbor. Given a sentence $c^{(1:n)}$ as input, a k-sized window will glide from character $c^{(1)}$ to $c^{(n)}$, where n represents the length of the sentence. k represents the window size, when k is 5; for each character $c^{(t)}\,(1 \le t \le n)$, the context characters $(c^{(t-2)}, c^{(t-1)}, c^{(t)}, c^{(t+1)}, c^{(t+2)})$ will be input into the lookup table layer for character matching. Characters beyond the boundary of the sentence are assigned to special symbols, specifically, the start and finish symbols. After that, the lookup table layer connects the character embedding to a single vector $x^{(t)} \in \mathbb{R}^{H_1}$, where $H_1 = k \times d$ represents the size of the first layer. Then $x^{(t)}$ is sent to the next layer, which transforms linearly. Finally, the sigmoid function $\sigma(x) = (1 + e^{-x})^{-1}$ is used as the activation function.

$$\begin{aligned} h^{(t)} &= \mathrm{g}\big(W_1 x^{(t)} + b_1\big), \\ y^{(t)} &= W_2 h^{(t)} + b_2, \end{aligned} \tag{2}$$

where $W_1 \in \mathbb{R}^{H_2 \times H_1}$, $b_1 \in \mathbb{R}^{H_2}$, $h^{(t)} \in \mathbb{R}^{H_2}$, $W_2 \in \mathbb{R}^{|\mathrm{T}| \times H_2}$, $b_2 \in \mathbb{R}^{|\mathrm{T}|}$, $y^{(t)} \in \mathbb{R}^{|\mathrm{T}|}$. H_2 represents the number of hidden units in the second layer, which exists as a hyperparameter. Given a set of tags T of size |T|, a linear transformation is carried out in a similar way, but it does not follow a nonlinear function. In Vietnamese word segmentation, the most commonly used tag set T is the character sequence set {B, M, E, S}.

In order to map the dependency between tags, we introduce the transition score A_{ij} proposed by Collobert et al. [29], which is used to measure the probability from the label $i \in \mathrm{T}$ to the label $j \in \mathrm{T}$. The transition score method can stably complete sequence tagging tasks such as Vietnamese word segmentation. The only disadvantage is that it can only use the context information of a limited-size window, and large-distance information may be ignored. The general architecture of the detailed Vietnamese word segmentation neural model is shown in Figure 5. It is mainly composed of three levels: the first layer is the character embedding layer, which inputs characters, then performs sequence tagging of the characters, and embeds morphemes; the second layer is a series of neural network layers, which performs feature learning and classification of the characters; and the last layer is the tag label inference layer, which performs label matching and inference on the predicted word segmentation characters. In particular, we adopt a bilinear structure, which examines the interaction of semantic information between different dimensions by computing the outer product of convolutional description vectors. Since different dimensions of the description vector correspond to different channels of convolutional features, and different channels extract different semantic features, the relationship between different semantic features of the input image can be captured simultaneously through bilinear operations.

3.4. Improved LSTM Model of Vietnamese Word Segmentation.

The initial stage in processing symbolic input with neural networks is to represent it as distributed vectors; it is also known as embeddings [30, 31]. In the Vietnamese word segmentation task, we manually built a character dictionary. Extract a character dictionary from the training set, and map unknown characters to different morpheme characters. A real-valued vector is used to represent each character. A vector matrix is created by stacking the embedded characters. The lookup database retrieves the corresponding characters after embedding. The lookup surface layer is the projection layer, and each embedded context morpheme character is implemented according to its index through a lookup table operation.

The long short-term memory neural network (LSTM) was presented in 1997 by Hochreiter and Schmidhuber [32]. LSTM is a derivative of the recurrent neural network (RNN). Since 2010, it has been demonstrated that RNNs have been successfully applied to speech recognition [33], language modeling [34], and text generation [35]. However, the disappearance of gradients and explosions makes RNN difficult to apply to long-term dynamics research. As an improved network of RNN, LSTM can handle this problem well. LSTM gives the network a great degree of freedom so that the network memory unit has an adaptive solution for learning and updating information. Therefore, LSTM neural network has an excellent performance in word segmentation tasks. The principle of the LSTM network is shown in Figure 6.

Assume that $X = (x_1, x_2, \ldots, x_n)$ represents an input sentence composed of word representations of n words. In every position t, the RNN produces a hidden layer h in the middle denoted as y_t, and the hidden state h_t uses a nonlinear activation function to update the previous hidden layers h_{t-1} and the input x_t, as shown below:

$$\begin{aligned} y_t &= \sigma\left(W_y h_t + b_y\right), \\ h_t &= f\left(h_{t-1}, x_t\right), \end{aligned} \tag{3}$$

where W_y and b_y are the parameter matrices and vectors learned during the training process, and σ represents the elementwise softmax function.

The LSTM unit includes an input gate i_t, a forget gate f_t, an output gate o_t, and a memory unit c_t to update the hidden state h_t, as shown below:

$$\begin{aligned} i_t &= \sigma\left(W_i x_t + V_i h_{t-1} + b_i\right), \\ f_t &= \sigma\left(W_f x_t + V_f h_{t-1} + b_f\right), \\ o_t &= \sigma\left(W_o x_t + V_o h_{t-1} + b_o\right), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh\left(W_c x_t + V_c h_{t-1} + b_c\right), \\ h_t &= o_t \odot \tanh\left(c_t\right), \end{aligned} \tag{4}$$

where $\odot$ is a kind of function which is similar to the multiplication operation, V represents a matrix related to weight, and b represents the learning vector. In order to improve the performance of the model, morpheme training was carried out on two LSTMs. The first one is a morpheme that begins on the left and works its way to the right; the next one is a reverse duplicate of a character. Before passing to the next layer, the outputs of the forward and reverse passes are combined in series. Finally, the activation function is used to obtain the prediction result.

In order to train the model, this paper takes advantage of pretrained word embeddings gathered through news website crawlers. In order to deal with the out-of-vocabulary (OOV) issues, we use a pretrained word embedding method and apply character-stage embedding from words. The character embedding is learned using the complete network after being randomly initialized. The overall architecture is shown in Figure 7. On the left side of the word segmentation, the forward LSTM estimates the description of the context, and the second LSTM calculates the morpheme character in the reverse direction and reads the sequence. All the representations are connected and linearly projected to the next layer, the size of which is the same as the amount of different word segmentation. Then we use the CRFs method to consider neighboring tags to generate the final context prediction for each word, in order to handle local information instead of the entire sentence in the long sentence domain, so as to enhance precision. Finally, in order to extract the local information between adjacent segments of the target word, we add an adaptive CNN layer at the end of the network.

4. Experiments

4.1. Datasets. Vietnamese is a scarce resource, and there is currently no publicly available dataset of Vietnamese subwords. For verifying the effectiveness of the method, we apply crawler technology to crawl Vietnamese news text data from the Vietnam Daily News website, use the word segmentation tool proposed by VnTokenizer to preprocess the crawled Vietnamese text data, and then obtain it through manual proofreading of 100,000 Vietnamese subword datasets. The process of making Vietnamese corpus dataset is shown in Figure 8. All the following experiments are based on this dataset for training. The test dataset contains 10,000 data and the training dataset contains 90,000 data. At the same time, based on constructing the word corpus, we further segment the vocabulary into syllables. All datasets are preprocessed by replacing Vietnamese corpus, continuous English characters, and numbers with unique signs. We use traditional scoring methods to calculate accuracy, recall, and F1 scores to evaluate the effect of Vietnamese word segmentation.

4.2. Training. All experiments in this paper are executed on the PyTorch framework and configure python 3.5 as the language environment version. The experimental hardware environment uses GTX 1080 Ti GPU, Intel i7-7700 CPU, 50 GB memory, and Pycharm Community 2020.3.2 as a development tool. Use the Max-Margin criterion during training. It concentrates on the model's decision boundary's robustness and proposes a probabilistic-based estimating approach as an alternative. So as to avoid the difficulty of overfitting during the training of the deep network, the dropout method presented by Srivastava et al. is also used in the training. The relevant training parameters are shown in Table 3.

4.3. Evaluation Metrics. In the experimental verification process, we choose precision, recall, and $F1$ score to evaluate the effectiveness of the method. Most classification and sequence labeling problems also adopt the above three evaluation factors. The equation is as follows:

$$\begin{aligned} F1 &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} \times \text{recall}}, \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}. \end{aligned} \tag{5}$$

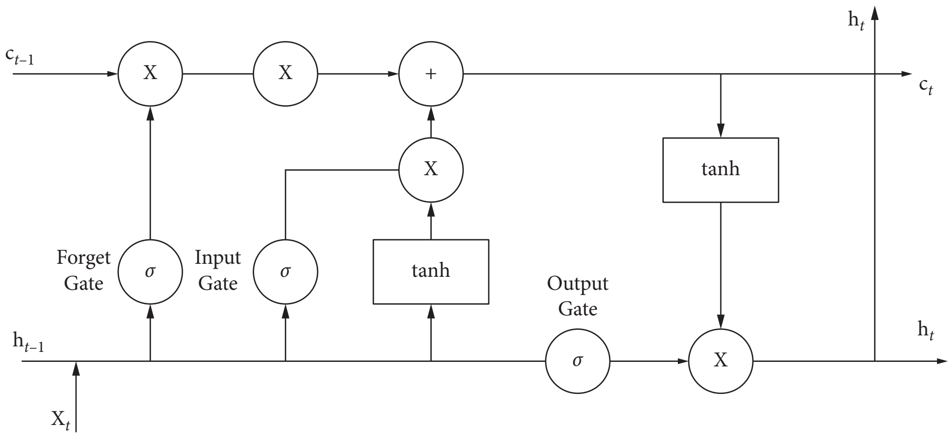

Figure 6: LSTM network function principle.

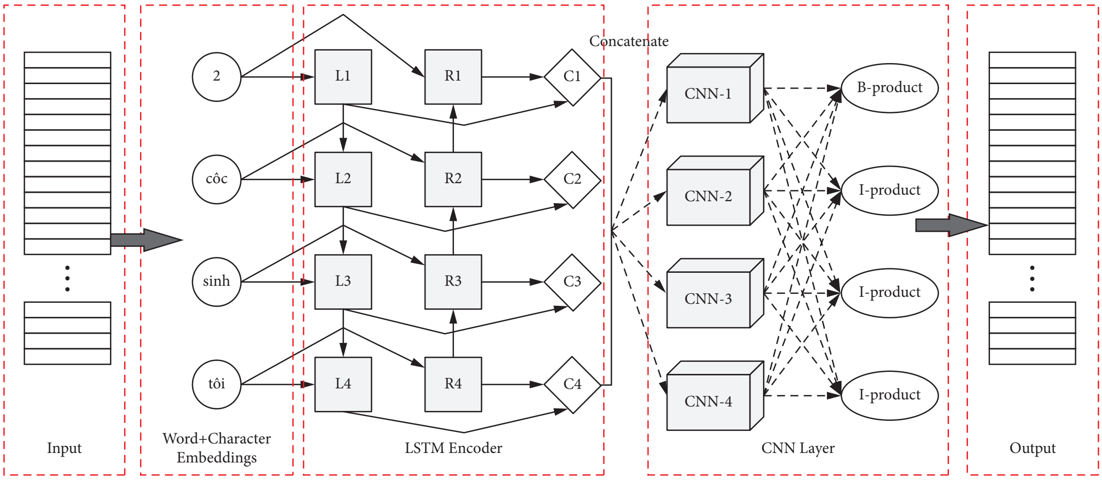

Figure 7: A framework of using improved LSTM for Vietnamese word segmentation.

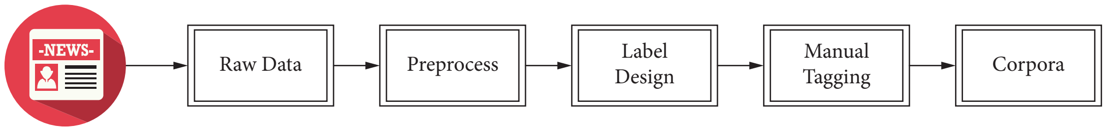

Figure 8: A framework of using improved LSTM for Vietnamese word segmentation.

Table 3: Training parameter settings.

Parameter	Value
Learning rate	0.15
Epoch	100
Context length	(−2,2)
Character embedding size	100
Regularization	0.0001
Margin loss discount	0.2
Dropout rate	0.5
Initial learning rate	0.2
Hidden unit number	200

In the above equation, TP (True Positive) is the number of properly recognized word segments. FP (False Positive) is the number of misrecognized word segments. FN (False Negative) indicates the number of unrecognized word segments.

4.4. Experimental Result. To verify how varying window widths affected the experimental results, we, respectively, compared the word segmentation effects under input sequence windows of different sizes. The comparison results are shown in Table 4.

Table 4: The influence of different window sizes on experimental results.

Window size	P	R	F
3	92.2	90.3	91.1
5	96.6	95.2	96.3
7	93.5	92.4	94.2
9	93.1	91.5	93.6

Table 5: The influence of different numbers of training corpora on experimental results.

Number of corpora	E1	E2
1	78.3	81.2
3	84.6	89.9
6	87.3	91.1
9	89.2	96.6

Table 4 shows that the choice of window size has an impact on the result of word segmentation, and the experimental results are normally distributed. Among the common Vietnamese words, most Vietnamese words contain 2 to 5 syllables. When the window is 5, the experimental results are the best, which can not only avoid the current syllable's insufficient dependence on the upper and lower syllables but also avoid the excessively long window. The syllable features are redundant. Therefore, the syllable window in this experiment is set to 5.

For confirming the impact of varied training corpus scales on the experimental results and to test the influence of the Vietnamese syllable structure knowledge feature on the character embedding vector representation effect, we conducted a comparative experiment on different scales of the training corpus. The results are shown in Table 5. Among them, E1 represents the word segmentation accuracy of the method under different corpus scales, and E2 represents the word segmentation accuracy of the method under the same corpus scale.

Table 5 shows that when the corpus is small, more features can be provided during neural network training. When the corpus increased from 10,000 to 30,000, the accuracy of Vietnamese word segmentation was significantly improved. The main reason was that the problem of insufficient feature coverage in the case of the small-scale corpus was supplemented, which caused the deep network to be unable to learn more features. At the same time, the method in this paper is faster than the CNN network in terms of accuracy rate improvement and combines the representation method of the syllable structure knowledge vector. In the case of the same corpus, it can provide more special segmentation morphemes for neural network learning. When the number of corpora was increased from 60,000 to 90,000, the accuracy of Vietnamese word segmentation was increased by 5.5%. It can be seen that the integration of linguistic knowledge features in the Vietnamese word segmentation vector representation can effectively compensate for the impact of resource scarcity on the performance of Vietnamese word segmentation. Therefore, the number of training corpus in the final experimental verification of this paper is 90,000.

Previous work found that by pretraining character embedding on unlabeled data, the model's overall performance can be improved. In the pretraining character embedding work in the homemade Vietnamese corpus, we used the toolkit researched by Mikolov et al. [36]. The character search database is initialized with the acquired embedding character, which replaces the earlier random initialization process. In this section, we verify it on a self-made dataset and compare it with the most advanced methods MaxEnt, biLSTM, and CNN. The experimental results are shown in Table 6.

Table 6: Comparison of the performance of different methods on Vietnamese word segmentation.

	P	R	F
MaxEnt	84.3	82.5	82.9
biLSTM	92.6	90.1	91.5
CNN	91.3	88.2	90.7
Ours	96.6	95.2	96.3

The experimental results show that, comparing with the CNN method, the accuracy of our method is boosted by 5.3%. Comparing with the biLSTM method, the accuracy of our method is boosted by 4%. Comparing with the MaxEnt method, the accuracy of our method is boosted by 12.3%. Regardless of whether it is from the recall rate or F1 score, the performance of the method proposed in this paper is even better, which proves the effectiveness of the method in this paper.

5. Conclusion

In this paper, we analyze the related work and research status of word segmentation in language processing and then lead to the study of Vietnamese word segmentation with a sparse corpus. The problem of combinational ambiguity and cross ambiguity in Vietnamese poses a great challenge to the task of Vietnamese word segmentation. In order to realize the Vietnamese word segmentation task, an improved LSTM neural network framework is proposed, and the Vietnamese word segmentation task is separated into a classification part and a sequence labeling part. We abandoned the traditional methods and chose the most advanced deep neural network to gain beneficial features at the character level. In order to test the method's performance, a verification experiment was conducted through a self-made Vietnamese word corpus. The experimental results show that, overall, the performance of the Vietnamese word separation system can be markedly increased by applying neural networks. In general, the accuracy of our method in the Vietnamese word segmentation task reached 96.6%, the recall rate reached 95.2%, and the F1 value reached 96.3%, which is significantly better than the traditional methods, CNN and LSTM methods.

In the future, we hope to use bidirectional recurrent neural networks to process sequences in two directions. Some adjustments are made to the method so that it can work well in other Southeast Asian language fields. Research on deeper semantic feature extraction may also be another direction in the future. In addition, self-made datasets have imbalance problems; in our future research, more attention will be paid to the construction and balance of the dataset.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Ji, Z. Lu, and H. Li, “An information retrieval approach to short text conversation,” 2014, https://arxiv.org/abs/1408.6988.

[2] Z. Yan, N. Duan, J. Bao et al., “Docchat: an information retrieval approach for chatbot engines using unstructured documents,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 516–525, Berlin, Germany, January 2016.

[3] D. A. Ali and N. Habash, “Botta: an Arabic dialect chatbot,” in *Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pp. 208–212, Osaka, Japan, December 2016.

[4] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, “Superagent: a customer service chatbot for e-commerce websites,” in *Proceedings of the ACL 2017, System Demonstrations*, pp. 97–102, Vancouver, Canada, 2017.

[5] C. Li, L. Li, and J. Qi, “A self-attentive model with gate mechanism for spoken language understanding,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3824–3833, USA, January 2018.

[6] Z. Yan, N. Duan, P. Chen, M. Zhou, J. Zhou, and Z. Li, “Building task-oriented dialogue systems for online shopping,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco California, USA, February 2017.

[7] T.-L. Ngo, V.-H. Nguyen, T.-H.-Y. Vuong et al., “Identifying user intents in Vietnamese spoken language commands and its application in smart mobile voice interaction,” in *Proceedings of the Asian Conference on Intelligent Information and Database Systems*, pp. 190–201, Springer, Da Nang, Vietnam, March 2016.

[8] P.-N. Tran, V.-D. Ta, Q.-T. Truong, Q.-V. Duong, T.-T. Nguyen, and X.-H. Phan, “Named entity recognition for Vietnamese spoken texts and its application in smart mobile voice interaction,” in *Proceedings of the Asian Conference on Intelligent Information and Database Systems*, pp. 170–180, Springer, Da Nang, Vietnam, March 2016.

[9] N. T. M. Huyen, A. Roussanaly, and H. T. Vinh, “A hybrid approach to word segmentation of Vietnamese texts,” in *Proceedings of the International conference on language and automata theory and applications*, pp. 240–249, Springer, Tarragona, Spain, March 2008.

[10] H. Tseng, P. C. Chang, G. Andrew, and D. Jurafsky, “A conditional random field word segmenter for sighan bakeoff 2005,” in *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*, Jeju Island, Korea, January 2005.

[11] N. X. Bach, N. D. Linh, and T. M. Phuong, “An empirical study on POS tagging for Vietnamese social media text,” *Computer Speech & Language*, vol. 50, pp. 1–15, 2018.

[12] J. Hu, G. Wang, F. Lochovsky, J. T. Sun, and Z. Chen, “Understanding user’s query intent with wikipedia,” in *Proceedings of the 18th international conference on World wide web*, pp. 471–480, Madrid Spain, April 2009.

[13] M. Mendoza and J. Zamora, “Identifying the intent of a user query using support vector machines,” in *Proceedings of the International symposium on string processing and information retrieval*, pp. 131–142, Springer, Saariselkä, Finland, Auguest 2009.

[14] Y. Shi, K. Yao, L. Tian, and D. Jiang, “Deep LSTM based feature mapping for query classification,” in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 1501–1511, San Diego CA, USA, January 2016.

[15] T. Kato, A. Nagai, N. Noda, R. Sumitomo, J. Wu, and S. Yamamoto, “Utterance intent classification of a spoken dialogue system with efficiently untied recursive autoencoders,” in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 60–64, Germany, Auguest 2017.

[16] C. W. Goo, G. Gao, Y. K. Hsu et al., “Slot-gated modeling for joint slot filling and intent prediction,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 753–757, Louisiana, NO, USA, January 2018.

[17] M. Qiu, F. L. Li, S. Wang et al., “Alime chat: a sequence to sequence and rerank based chatbot engine,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 498–503Short Papers), Vancouver, Canada, January 2017.

[18] P. S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, “Learning deep structured semantic models for web search using clickthrough data,” in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 2333–2338, San Francisco California USA, October 2013.

[19] N. X. Bach, K. Hiraishi, N. Le Minh, and A. Shimazu, “Dual Decomposition for Vietnamese part-of-speech tagging,” *Procedia Computer Science*, vol. 22, pp. 123–131, 2013.

[20] D. Q. Nguyen, D. Q. Nguyen, S. B. Pham, P.-T. Nguyen, and M. Le Nguyen, “From treebank conversion to automatic dependency parsing for Vietnamese,” in *Proceedings of the International Conference on Applications of Natural Language to Data Bases/Information Systems*, pp. 196–207, Springer, Montpellier, France, June 2014.

[21] Y. Li, J. Y. Guo, Z. T. Yu, C. Mao, and Y. Xian, “Constituent-to-Dependency conversion for Vietnamese[J],” *Journal of Frontiers of Computer Science and Technology*, vol. 11, no. 4, pp. 599–607, 2017.

[22] H. Nguyễn Đ, M. J. Alves, and H. C. Nguyễn, *Vietnamese[M]*, Routledge, England, UK, 2018.

[23] L. C. A. Thompson, *Vietnamese Grammar*, University of Hawaii Press, Honolulu, HI, USA, 1988.

[24] L. A. Michaelis, “Expectation contravention and use ambiguity: the Vietnamese connective cũng,” *Journal of Pragmatics*, vol. 21, no. 1, pp. 1–36, 1994.

[25] H. V. Luong, “Plural markers and personal pronouns in Vietnamese person reference: an analysis of pragmatic ambiguity and native models,” *Anthropological Linguistics*, pp. 49–70, 1987.

[26] C. Miller, “Structural ambiguity in the Vietnamese relative clause,” *Mon-Khmer Studies*, vol. 5, pp. 233–267, 1976.

[27] B. Tesar, “Enforcing grammatical restrictiveness can help resolve structural ambiguity,” vol. 21, pp. 443–456, in *Proceedings of the West Coast Conference on Formal Linguistics*, vol. 21, pp. 443–456, Cascadilla Press, USA, April 2002.

[28] M. S. Zhang, Z. L. Deng, W. X. Che, and T. Liu, "Combining statistical model and dictionary for domain adaption of Chinese word segmentation," *Journal of Chinese Information*, vol. 26, no. 2, pp. 8–13, 2012.

[29] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. ARTICLE, pp. 2493–2537, 2011.

[30] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, Helsinki Finland, July 2008.

[31] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] O. Vinyals, S. V. Ravuri, and D Povey, ". Revisiting recurrent neural networks for robust ASR," in *Proceedings of the 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4085–4088, IEEE, Kyoto, Japan, March 2012.

[34] T. Mikolov, M. Karafiát, L. Burget, J. Cernocky, and S. Khudapur, "Recurrent neural network based language model," in *Proceedings of the Interspeech Communication Association*, vol. 2, no. 3, pp. 1045–1048, chiba, Japan, September 2010.

[35] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proceedings of the International conference on Machine Learnung ICML*, Washington, USA, July 2011.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.