Thai Word Segmentation Based on Sequence-to-Sequence Model

Huiwen Wu, Yongzhong Huang*, Cancheng Li, Haoyu Zhuang, Xinyu Xu
School of Computer Science and Information Security, Guilin University of Electronic Technology, No.1, Jinji Road, Guilin 541004, China

*wuhuiwen12@126.com

Abstract. Thai as a low-resource language has a large word segmentation performance improvement space. In this paper, we investigate a sequence-to-sequence model for Thai word segmentation with two different recurrent neural networks, which could transform one input sequence into another output sequence. Furthermore, we evaluate datasets in four different fields compared then with other multiple word segmentation models, and the F1 value in the encyclopedia dataset reaches 97.15%. The results show that the proposed model has superior performance and is more effective, it is worth mentioning that the expected results can be achieved even with limited data resources.

Keywords: Thai, Low-Resource, Word Segmentation, Sequence-to-Sequence.

1. Introduction

Word segmentation is a fundamental task, and its performance improvement is crucial for downstream tasks of natural language processing (NLP). Word segmentation is the process of recombining consecutive subword sequences into word sequences according to certain specifications. At present, deep learning has been widely used in various fields of NLP. Particularly, some research on word segmentation technology has once again received extensive attention. Li et al.[1] conducted verification in multiple NLP tasks, such as machine translation and text classification. It shows that under the premise of the current wide application of deep learning, it is still necessary to further improve the performance of word segmentation.

There is no clear separator between Thai words, which is similar to Chinese and belongs to the Tai Branch of the Kam-Tai Family, and word formation is more flexible[2]. At the present stage, the Chinese word segmentation (CWS) is relatively mature compare with Thai word segmentation (TWS). The main reason is that these minority languages in ASEAN countries are scarce-corpus and high-quality datasets are limited, which is difficult to provide sufficient data support. Therefore, the challenge of word segmentation will be greater. In this paper, we propose a model based on sequence-to-sequence (Seq2Seq)of the TWS combine the CWS and the characteristics of Thai word formation. We apply the Thai syllable structure features to word embedding and transform the word segmentation task into a sequence conversion task through the Seq2Seq model to realizing TWS.

The remainder of the paper is organized as follows. Related researches on word segmentation and some classic model algorithms are provided in section 2. The TWS model and the analysis of the experimental results proposed are described in section 3 and section 4 respectively. These are conclusions and outlooks at the end of this paper.
2. Related works
There are many studies on CWS. In the early days, there were more methods based on machine learning, such as maximum entropy (ME) [3] and conditional random field (CRF) [4]. In view of the current powerful feature learning ability of deep learning, more and more researchers apply deep learning to word segmentation. Zhang et al. [5] applied deep neural networks to CWS and used the perceptron-style algorithm for training, reducing model training time. But the general neural network cannot learn long-distance sentence information well. Chen et al. [6] proposed the CWS method of long short term memory (LSTM) neural network that could better learn long-distance sentence information. The word segmentation method of LSTM ignores the sentence information from right-to-left. Yao et al. [7] aimed at this problem and adopted the bidirectional long short term memory (Bi-LSTM) model to achieve CWS, which can also consider context information to obtain better word segmentation results. In 2015, Huang et al. [8] combined CRF into the Bi-LSTM model using the advantages of Bi-LSTM and combining the CRF layer to effectively use sentence-level tag information. After that, it has been widely used in word segmentation, with stronger robustness and better model performance.

At this stage, there are few studies on TWS, and it is more complicated than CWS. In [9], Aroonmanakun segmented according to Thai syllables and merged them into words using the strength of the collocation between the dictionary and the syllables. Bheganan et al. [10] combined dictionary and non-dictionary word segmentation methods, using hidden Markov and decision tree methods to achieve TWS, which somewhat improved the word segmentation. However, these methods of combining dictionaries depend on the quality and quantity of the dictionary, further, there will be certain errors in the segmentation of unregistered words. Haruchaiyasak et al. [11] regarded word segmentation as a sequence tagging task, using the CRF model and abandoning the dictionary method. It reduces the workload of the dictionary method and performs additional processing on named entities, thereby improving the effect of the model. Zhao [12] also used the CRF model to study TWS, but he used less data. In recent years, deep learning has also been applied to TWS. Similarly, the multi-layer neural network for TWS in response to the complex feature templates and large search space of traditional TWS is applied in [13]. Compared with traditional machine learning methods, word segmentation with better performance. In the case of sufficient corpus, the multi-layer neural network segmentation method performs better. But minority language corpora are scarce, and the word segmentation model needs to be more effective.

Consequently, responding to the lack of corpus in minority languages, a more effective Seq2Seq Thai work segmentation method is proposed. We conducted experiments on datasets in four different fields and compared them with GRU, GRU-CRF, Bi-LSTM, and Bi-LSTM-CRF models. The experiments have shown that our proposed method is more effective than others and is more suitable for TWS when the corpus of the minority languages is limited.

3. Model
In 2014, Cho [14] and Sutskever [15] first proposed the Seq2Seq model, which achieved performance results in the field of machine translation. Due to the strong versatility of the model and expansibility, it is widely used in multiple tasks of NLP, such as machine translation, automatic document summarization, and chat robots. All that is needed to complete a task is to convert it into another predicted symbol sequence. Therefore, we extend the traditional basic Seq2Seq two-layer model to a three-layer by introducing the Glove of the global vectors, so as to effectively realize the TWS task.

3.1. Model-based Seq2Seq
The basic Seq2Seq model is composed of the encoder and decoder. The encoder is used to analyze the input sequence and convert the variable-length input sequence into a fixed-length background variable \( C \), then encode the input sequence information in the \( C \). Assume that the input sentence is \( X = (x_1, x_2, \ldots, x_T) \), \( x_t \) represents the \( t \)-th word in the sentence. The hidden state \( h_t \) at
moment \( t \) is determined by the feature vector \( x_t' \) of the input \( x_t \) and the hidden state \( h_{t-1} \) at the previous moment. The function \( f \) represents the transformation of the hidden layer of the RNN as Eq. (1).

\[
h_t = f(x_t', h_{t-1}) \tag{1}
\]

The background variable in the middle is composed of the hidden state at each moment and transforms into the \( C \) through the function \( q \), which also serves as the initial state of the next decoder.

\[
C = q(h_1, h_2, ..., h_T). \tag{2}
\]

Where \( q(h_1, h_2, ..., h_T) = h_\tau \) is be selected, then the background variable is the hidden state \( h_\tau \) at the final moment of the input sequence.

The decoding process is just the opposite of the encoding process. The decoder will predict the next possible target value \( y_\tau \) based on the generated output sequence \( y_1, y_2, ..., y_{\tau-1} \) and the \( C \), as in \( P(y_\tau | y_1, ..., y_{\tau-1}, C) \). At the moment \( t' \) of the output sequence, the decoder takes the output \( y_{t'-1} \) at the previous moment and \( C \) as input, which transforms them and the hidden state \( s_{t'-1} \) at the previous moment into the hidden state \( s_t \) at the current moment. The function \( g \) represents the decoder hiding layer transformation, as in (3).

\[
s_t = g(y_{t'-1}, C, s_{t'-1}) \tag{3}
\]

After obtaining the corresponding hidden state and then calculate \( P(y_\tau | y_1, ..., y_{\tau-1}, C) \) through a custom output layer and softmax. The specific definition of the basic Seq2Seq model can be explained in more detail in [16].

3.2. **Glove-Seq2Seq word segmentation model**

Our model is mainly divided into embedding, encoder, and decoder layers. We introduce the external pretrain word embeddings through Glove. Then Bi-LSTM is used to encode the input sequence in the encoder layer, the encoding information is obtained and the decoder layer uses a gated recurrent unit (GRU) to decode. Finally, the tag information of the element is obtained based on the decoding sequence to realize TWS. The experimental model of this paper is shown in Fig. 1.

**Embedding layer.** We use Glove [17] pretrain word embedding. Although there are many pretrain language models like the Bert and XLNet. However, these models are relatively difficult to implement, and there is a lack of well-trained Thai pretrain language models. It is not the best choice for minority languages with the scarce corpus. The smallest word formation granularity in Thai is syllables instead of characters, which is different from Chinese. In the Thai writing system, consonants and vowels are combined according to rules and then combined with tones to form a syllable. Only when the syllables are combined with a specific meaning can form a language unit [2]. Therefore, according to the characteristics of Thai word formation, we use syllable-based word embedding as the initial input of the model.
**Encoder layer.** After introducing syllable-based word embedding, the encoder layer encodes the input sequence information through Bi-LSTM. When segmenting a word in the input sentence, it is often necessary to segment the previous word according to the following word information. Therefore, in the input sequence, the sequence information of future words is also particularly important. The Bi-LSTM of we use can consider both the left-to-right and right-to-left sequence information in the sentence, it avoids the problem only considers the sequence information in a single direction, which improves the effectiveness of the model.

The encoder layer obtains two sequence features from left-to-right $x_1, x_2, \ldots, x_T$ and right-to-left $x_1, x_2, \ldots, x_T$ through two LSTMs in different directions. Then the hidden state output of the encoder is

$$h_t = h_t^{\text{left}} \oplus h_t^{\text{right}} (t \in [1, T])$$

$h_t^{\text{left}}$ and $h_t^{\text{right}}$ are the hidden states of the encoder from left-to-right and right-to-left, respectively. Taking the LSTM from left-to-right as an example, the input at moment $t$ is $x_t$, which calculation of the forget gate $f_t$, input gate $i_t$, and output gate $o_t$ are as in (4-9).

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

(4)

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

(5)

$$c_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

(6)
\[
\begin{align*}
\overrightarrow{c}_t &= \overrightarrow{f}_t \overrightarrow{c}_{t-1} + \overrightarrow{i}_t \overrightarrow{\tilde{c}_t}, \\
\overrightarrow{o}_t &= \sigma(\overrightarrow{W}_o \overrightarrow{x}_t + \overrightarrow{U}_o \overrightarrow{h}_{t-1} + \overrightarrow{b}_o), \\
\overrightarrow{h}_t &= \overrightarrow{o}_t \tanh(\overrightarrow{c}_t).
\end{align*}
\] (7)
(8)
(9)

Among them, \( \overrightarrow{c}_t \) and \( \overrightarrow{\tilde{c}_t} \) represent temporary memory cells and memory cells that have been updated at moment \( t \), respectively. The parameters of \( \overrightarrow{W} \) and \( \overrightarrow{U} \) represent different weight parameters. \( \overrightarrow{b} \) is a deviation parameter and \( \overrightarrow{h}_t \) represents the hidden state at moment \( t \).

**Decoder layer.** The decoder layer is to predict the next possible output sequence \( y'_{t'} \) based on the output sequence \( y_1, y_2, \ldots, y_{t'-1} \) at the previous moment and the \( C \). In our Seq2Seq Thai word segmentation model, the attention mechanism is omitted, and we use the \( t' \)-th hidden vector of the encoding output in the time step \( t' \). In the commonly used Seq2Seq model, an attention mechanism is usually added to assign different weights to the encoded information at different moments in the input sequence. However, in the sequence conversion process of our model, the next step output is directly defined at a certain input position. As shown in Tab 1, the next input in Step 1 corresponds to the second syllable in the sentence. And so on, the input at each step corresponds to the syllable in the input sequence. More precisely, our attention is clearly defined by the accompanying transition state during the decoding process of the model, as mentioned by Mei et al. [18].

We used the GRU with a simpler structure and easier to train as the sequence output. Not only can effectively reduce the model decoding time, but also not easy to overfit, which further improves the effectiveness of the model. The calculation of the reset gate \( \overrightarrow{Z}_{t'} \) and update gate \( \overrightarrow{R}_{t'} \) are as in (10-13).

\[
\begin{align*}
\overrightarrow{R}_{t'} &= \sigma(\overrightarrow{W}_r \overrightarrow{x}_{t'} + \overrightarrow{U}_r \overrightarrow{h}_{t'-1} + \overrightarrow{b}_r), \\
\overrightarrow{Z}_{t'} &= \sigma(\overrightarrow{W}_z \overrightarrow{x}_{t'} + \overrightarrow{U}_z \overrightarrow{h}_{t'-1} + \overrightarrow{b}_z), \\
\overrightarrow{\tilde{h}_{t'}} &= \tanh(\overrightarrow{W}_h \overrightarrow{x}_{t'} + (\overrightarrow{R}_{t'} \overrightarrow{h}_{t'-1})\overrightarrow{U}_h + \overrightarrow{b}_h), \\
\overrightarrow{h}_{t'} &= \overrightarrow{Z}_{t'} \overrightarrow{h}_{t'-1} + (1 - \overrightarrow{Z}_{t'}) \overrightarrow{\tilde{h}_{t'}}.
\end{align*}
\] (10)
(11)
(12)
(13)

Among them, the parameters of \( \overrightarrow{W} \) and \( \overrightarrow{U} \) represent different weight parameters. \( \overrightarrow{b} \) is a deviation parameter. \( \overrightarrow{h}_t \) and \( \overrightarrow{\tilde{h}_{t'}} \) represent the hidden state and candidate hidden state at moment \( t' \), respectively.

**3.3. Training**

We used the cross-entropy loss to train our model. According to preliminary experiments, it is found that the performance of the model is stronger by using cross-entropy loss. In an input sequence, the loss function \( L \) is calculated as in (14).

\[
L = \frac{1}{T} \sum_t H(y_t, p_t) = -\frac{1}{T} \sum_t \sum_i y_{ti} \log(p_{ti}).
\] (14)
Among them, $i$ represents the number of sequence labels, $y$ and $p$ represents the true sequence label probability and predicted probability of the model after training, respectively.

Where we tried to use the AdamW optimizer[19], which uses decoupling weight decay and has better generalization performance than Adam [20]. In this paper, we also present the SGD, Adadelta, and Adam optimizers. We found that after using SGD and Adadelta optimizers, the model converges slowly. The AdamW has better performance and fast convergence compared with using Adam. We added the dropout [21] to the syllable-based word embedding including the encoder and decoder layers, which effectively reduced the model overfitting. The training process of the test set is shown in Fig. 2 and Fig. 3, respectively.

4. Experiments

4.1. Datasets

The data used in the experiment comes from InterBEST 2009 [22], which mainly contains datasets in four different fields i.e., namely article, news, encyclopedia, and novel fields. To better verify the effectiveness of the model, we randomly selected 10-20 .txt files from each of the four types of data for experiments. Furthermore, the data adopts the BMES labeling method[23]. Finally, We divide the data into training, development, and test sets in proportion as shown in Tab 2.

4.2. Results and analysis

To evaluate the proposed model, we use three popular evaluation indexes including the precision (P), recall (R), f1-score (F1), and F1 averages among four datasets. Our model compares the result of GRU, Bi-LSTM neural network word segmentation models, hybrid GRU-CRF, Bi-LSTM-CRF word segmentation models as shown in Tab 3 to Tab 7, respectively.

We combine two different neural network models into a word segmentation model structure in the form of encoder and decoder have given in Tab 7. By comparing it with Tab 3 and Tab 4, it can be seen that the Seq2Seq model is significantly better than the word segmentation model of GRU and Bi-LSTM neural network. Furthermore, compared with the GRU model, our model improves by nearly 5.5% on the F1 average of the four datasets, while the Bi-LSTM model improves by nearly 2.5%. It can also be seen that the Bi-LSTM has better performance than the word segmentation model of GRU. After combining CRF, the performance of the two models has been improved. Among them, GRU-CRF increased by 1.61% and Bi-LSTM-CRF increased by 1.29% as shown in Tab 5 and Tab 6, respectively. The Bi-LSTM-CRF model is the most effective among the four comparison models, but the Seq2Seq model is more prominent. It is also found through experiments that the results in the encyclopedia field are the best, followed by the article field, in which the news and novel fields are slightly worse. Meanwhile, the performance in the four comparison models is similar. The reason may be that there are more entities in the data of news and novel and longer entities will have a certain
influence on the experiment of word segmentation. Haruechayiasak[11] also mentioned the influence of entities on word segmentation. Compared with traditional machine learning methods [12], our proposed method has excellent performance when using relatively few datasets. Compared with the multi-layer neural network word segmentation model studied [13], the experimental results are greatly improved compared with other methods, but the amount of data used is large. In this paper, the Seq2Seq Thai word segmentation model shows better experimental results with fewer data.
Tab. 2 Statistics of datasets.

| Dataset | Train | Dev | Test |
|---------|-------|-----|------|
| article | Sentence 2065 | 590 | 294 |
| Word | 143168 | 39880 | 20286 |
| encyclopedia | Sentence 5398 | 1542 | 770 |
| Word | 138237 | 40040 | 19887 |
| news | Sentence 3524 | 1006 | 502 |
| Word | 136757 | 39063 | 19642 |
| novel | Sentence 4690 | 1340 | 668 |
| Word | 142343 | 39138 | 19474 |

Tab. 3 Evaluation results of different datasets on the gru.

| Dataset | P | R | F1 |
|---------|---|---|----|
| article | 90.36 | 92.04 | 91.19 |
| encyclopedia | 92.10 | 93.22 | 92.66 |
| news | 90.16 | 90.76 | 90.46 |
| novel | 89.29 | 90.40 | 89.85 |
| Average | 90.48 | 91.61 | 91.04 |

Tab. 4 Evaluation results of different datasets on the bi-lstm.

| Dataset | P | R | F1 |
|---------|---|---|----|
| article | 93.96 | 95.03 | 94.49 |
| encyclopedia | 95.26 | 96.12 | 95.69 |
| news | 93.74 | 94.29 | 94.01 |
| novel | 92.17 | 94.07 | 93.11 |
| Average | 93.78 | 94.88 | 94.33 |

4.3. Hyper-parameters

Tab 8 shows the value of main hyper-parameters for our models, we adjusted the hyper-parameters in detail. Fig. 2 shows the loss value of the Seq2Seq model during the training process of the test set. It could be found that when the model starts training, the loss values of the four types of data are all below 0.3. After nearly 50 epochs of training, it is getting closer and closer to 0, which could be seen that the robustness of the model is better. Further, Fig. 3 shows the F1 value of the test set during 200 epoch training. As shown in the Fig, the model converges speed quickly and the overfitting phenomenon is not obvious during the training process. After nearly 40 epoch, the model is close to convergence. At the same time, we found that the domain applicability of the model is also strong, the gap between the four types of data experimental results is small, and the data points of the F1 value are clustered relatively high.

| Parameters | Value |
|------------|-------|
| Embedding size | 100 |
| Dropout rate | 0.25 |
| Learning rate | 0.001 |

Tab. 8 Hyper-parameter settings.
5. Conclusion
Aiming at the task of Thai word segmentation, this paper presents a more effective sequence-to-sequence Thai word segmentation model. The model uses bidirectional long short-term memory and the gated recurrent unit neural network for encoding input and decoding output respectively. Moreover, it was evaluated on InterBEST 2009 Thai corpus and achieved optimal results on four datasets in different fields. The experimental results show that the model is simple, effective, and strong domain applicability, which is more suitable for minority languages where data are scarce. We found that entities have a certain influence on the effect of word segmentation. In the future, we will consider the word segmentation with entities in sentences.

Acknowledgments
This work was supported by the National Natural Science Foundation of China under grant no. 61866008 and Basic and Applied Basic Research Fund of Guangdong Province, China under grant no. 2019B1515120085.

References
[1] Xiao, L., Yuxian, M., Xiaofei, S., Qinghong, H., Arianna, Y. and Jiwei, L. 2019 Is Word Segmentation Necessary for Deep Learning of Chinese Representations Meeting of the association for computational linguistics pp 3242-3252
[2] Xiao, Rui, Pand Wenze B 2017 Thai Grammar (Beijing: Peking University Press)
[3] Jin, K. L., Hwee, T. N. and Wenyan, G. 2005 A Maximum Entropy Approach to Chinese Word Segmentation Proc. of the Fourth SIGHAN Workshop on Chinese Language Processing
[4] Hai Z., Changning H., Mu L. and Bao-Liang L 2006 Effective Tag Set Selection in Chinese Word Segmentation Via Conditional Random Field Modeling Proc. of the 20th Pacific Asia Conf. on Language, Information and Computation pp 87-94
[5] Xiaqing Z., Hanyang C. and Tianyu X. 2013 Deep Learning for Chinese Word Segmentation and Pos Tagging Proc. of the 2013 Conf. on Empirical Methods in NLP pp 647-657
[6] Xinchi C., Xipeng Q., Chexxi Z., Pengfei L. and Xuan-Jing H. 2015 Long Short-Term Memory Neural Networks for Chinese Word Segmentation Proc. of the 2015 Conf. on Empirical Methods in NLP pp 1197-1206
[7] Yushi Y. and Zheng H. 2016 Bi-Directional Lstm Recurrent Neural Network for Chinese Word Segmentation Int. Conf. on Neural Information Processing Springer pp 345-353
[8] Zhiheng H., Wei X. and Kai Y. 2015 Bidirectional Lstm-Crf Models for Sequence Tagging arXiv preprint arXiv:1508.01991
[9] Wirota A 2002 Collocation and Thai Word Segmentation Proc. Of SNLP-Oriental COCOSDA pp 68-75
[10] Poramin B, Richi N and Yue X 2009 Thai Word Segmentation with Hidden Markov Model and
Decision Tree Pacific-Asia Conf. on Knowledge Discovery and Data Mining Springer pp 74-85

[11] Choochart H and Sarawoot K 2009 Tlex: Thai Lexeme Analyser Based on the Conditional Random Fields Proc. of 8th Int. Symp. on NLP

[12] Zhipeng Z 2014 The Machine Learning Method Is Adopted to Realize Thai Word Segmentation Yunnan University pp 1-2

[13] Guang-feng T, Yan-tuan X, Hong-bin W and Shu-juan W 2018 A Context Character Feature Based Neural Network Model for Thai Word Segmentation Computer Engineering and Science p 26

[14] Kyunghyun C, Bart V M, Caglar G, Dzmitry B, Fethi B, Holger S and Yoshua B 2014 Learning Phrase Representations Using Rnn Encoder-Decoder for Statistical Machine Translation arXiv preprint arXiv:1406.1078

[15] Ilya Sutskever, Oriol Vinyals and Quoc V Le 2014 Sequence to Sequence Learning with Neural Networks Advances in Neural Information Processing Systems pp 3104-3112

[16] Aston Z, Zachary C. L, Mu L and Alexander J. S 2020 Dive into Deep Learning

[17] Jeffrey P, Richard S and Christopher D M 2014 Glove: Global Vectors for Word Representation Proc. of the 2014 Conf. on Empirical Methods in NLP (EMNLP) pp 1532-1543

[18] Meishan Z, Nan Y and Guohong F 2018 A Simple and Effective Neural Model for Joint Word Segmentation and Pos Tagging IEEE/ACM Transactions on Audio, Speech, and Language Processing 26 pp 1528-1538

[19] Ilya Loshchilov and Frank H 2017 Decoupled Weight Decay Regularization arXiv preprint arXiv:1711.05101

[20] Diederik P K and Jimmy B 2014 Adam: A Method for Stochastic Optimization arXiv preprint arXiv:1412.6980

[21] Nitish S, Geoffrey H, Alex K, Ilya S and Ruslan S 2014 Dropout: A Simple Way to Prevent Neural Networks from Overfitting J. The journal of machine learning research 15 pp 1929-1958

[22] Monthika B, Kanyanut K, Patcharika C, Sitthaa P, Sumonmas P, Tipraporn T and Krit K 2009 Best Corpus Development and Analysis 2009 Int. Conf. on Asian Language Processing IEEE pp 322-327

[23] Nianwen X 2003 Chinese Word Segmentation as Character Tagging Special Issue on Word Formation and Chinese Language Processing Int. Journal of Computational Linguistics & Chinese Language Processing 8 pp 29-48