Incorporating Inner-word and Out-word Features for Mongolian Morphological Segmentation

Na Liu[1,2,3], **Xiangdong Su**[1,2,*], **Haoran Zhang**[1,2], **Guanglai Gao**[1,2], **Feilong Bao**[1,2]
[1]Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, China
[2]College of Computer Science Inner Mongolia University, China
[3]Department of Mathematics and Computer Science Hetao University, China
cssxd@imu.edu.cn

Abstract

Mongolian morphological segmentation is regarded as a crucial preprocessing step in many Mongolian related NLP applications and has received extensive attention. Recently, end-to-end segmentation approaches with long short-term memory networks (LSTM) have achieved excellent results. However, the inner-word features among characters in the word and the out-word features from context are not well utilized in the segmentation process. In this paper, we propose a neural network incorporating inner-word and out-word features for Mongolian morphological segmentation. The network consists of two encoders and one decoder. The inner-word encoder uses the self-attention mechanisms to capture the inner-word features of the target word. The out-word encoder employs a two layers BiLSTM network to extract out-word features in the sentence. Then, the decoder adopts a multi-head double attention layer to fuse the inner-word features and out-word features and produces the segmentation result. The evaluation experiment compares the proposed network with the baselines and explores the effectiveness of the sub-modules.

1 Introduction

Mongolian is a morphologically rich language and its words are formed by attaching suffixes to roots (Kullmann and Tserenpil, 2008). Each word has a root and zero or more suffixes, which are called Mongolian morphemes. The morphemes in a word indicate the basic word features and provide grammatical and semantic relations among words in the sentence. Mongolian morphological segmentation aims to split Mongolian words into their morphemes, which facilitates the Mongolian NLP tasks, such as name entity recognition (Wang et al., 2016; Wang et al., 2019), information retrieval (Liu et al., 2012), machine translation (Fan et al., 2017; Yang et al., 2016), and speech synthesis (Liu et al., 2017). There are about 60 thousand of morphemes in Mongolian, and the number of their formed words is more than 7 million. It becomes a tendency to process Mongolian text on morphemes rather than on words to make full use of the morpheme information in Mongolian NLP tasks. Besides, Segmenting Mongolian words into morpheme can alleviate the data-sparse problem and out-of-vocabulary (OOV) problem. Therefore, Mongolian morphological segmentation is an essential preprocessing step and effects the downstream Mongolian NLP tasks.

Mongolian morphological segmentation is closely related to the words themselves and their context. Table 1 shows several morphological segmentation examples. In usual, a Mongolian word corresponds to only one segmentation, such as the target words "ᠪᠠᠷᠢᠪᠠ (bariba)" and "ᠪᠠᠷᠢᠭᠤᠯᠪᠠ (barigulba)" in sentences I and II. But in some cases, parts of Mongolian words correspond to different segmentation results according to the context where they appear. For example, the target word "ᠬᠢᠭᠡᠳ (higed)" in the sentences V and VI is segmented into different morphemes due to its different contexts. Such words are called multi-category word. In addition, some morphemes are the constituent parts of other morphemes. This further makes the segmentation more difficult. Here, the unit "ᠨ (n)" in the word "ᠨᠡᠭᠤᠨ (negun)" in the sentence III is a morpheme while it is just part of the morpheme "ᠬᠠᠨ (han)" in the word "ᠠᠯᠭᠤᠷᠪᠠᠨ (algvrhan)" in the sentence IV. In summary, Mongolian morphological segmentation is still a challenging task.

Proceedings of the 28th International Conference on Computational Linguistics, pages 4638–4648
Barcelona, Spain (Online), December 8-13, 2020

Sentence	Target word and its segmentation	Meaning of the sentence
I. ᠲᠡᠷᠡ ᠪᠠᠢᠰᠢᠨ᠊ ᠪᠠᠷᠢᠪᠠ ᠃᠃ (tere baising bariba.)	ᠪᠠᠷᠢᠪᠠ (bariba) ᠪᠠᠷᠢ+ᠪᠠ (bari+ba)	*He builts the house.*
II. ᠲᠡᠷᠡ ᠪᠠᠢᠰᠢᠨ᠊ ᠪᠠᠷᠢᠭᠤᠯᠪᠠ ᠃᠃ (tere baising barigulba.)	ᠪᠠᠷᠢᠭᠤᠯᠪᠠ (barigulba) ᠪᠠᠷᠢ+ᠭᠤᠯ+ᠪᠠ (bari+gul+ba)	*He had the house built.*
III. ᠲᠡᠷᠡ ᠲᠤᠰ ᠭᠠᠵᠠᠷ ᠨᠡᠭᠦᠨ᠊ ᠰᠠᠭᠤᠷᠪᠢᠬᠡᠪᠠ ᠃᠃ (tere tvs gajar negun sagvrixiba .)	ᠨᠡᠭᠦᠨ (negun) ᠨᠡᠭᠦ+ᠨ (negu+n)	*He emigrated to the local area.*
IV. ᠲᠡᠷᠡ ᠠᠯᠭᠤᠷᠬᠠᠨ᠊ ᠭᠠᠵᠠᠷᠴᠤ ᠶᠠᠪᠤᠬᠤ ᠃ (tere algvrhan garcv yabvn_a.)	ᠠᠯᠭᠤᠷᠬᠠᠨ (algvrhan) ᠠᠯᠭᠤᠷ+ᠬᠠᠨ (algvr+han)	*He trudged out.*
V. ᠲᠡᠷᠡ ᠠᠵᠢᠯ ᠬᠢᠭᠡᠳ᠊ ᠶᠠᠪᠪᠠ ᠃᠃ (tere ajil higed yabvba.)	ᠬᠢᠭᠡᠳ (higed) ᠬᠢ+ᠭᠡᠳ (hi+ged)	*He went to work.*
VI. ᠲᠡᠷᠡ ᠨᠤᠮ ᠬᠢᠭᠡᠳ ᠬᠠᠷᠠᠨᠳᠠ ᠵᠢ ᠠᠪᠪᠠ ᠃᠃ (tere nqm higed haranda-ji abvba.)	ᠬᠢᠭᠡᠳ (higed) ᠬᠢᠭᠡᠳ (higed)	*He took the book and pencil.*

Table 1: An example of traditional Mongolian word segmentation and the words in the brackets is the Latin form of Mongolian words.

This study proposes a novel approach to Mongolian morphological segmentation, which addresses challenges by incorporating inner-word and out-word features. The proposed network consists of two encoders and one decoder. First, a standard self-attention network is utilized as an inner-word encoder, which conducts connections between two arbitrary characters in a word and draws the inner-word features directly. Meanwhile, a bidirectional LSTM (BiLSTM) network is used as the out-word encoder to extract the out-word features of the word in the sentence. Finally, a doubly attentive decoder is employed to fuse the inner-word features and out-word features and produce the segmentation result. The evaluation experiment compares the proposed network with the baselines and explores the effectiveness of the sub-modules.

The contribution of this paper is as follows. This paper proposed a network for Mongolian morphological segmentation according to the Mongolian characteristics. Two well-designed encoders were introduced in the network to extract the inner-word-level feature information and the out-word-level information between the target word and other words in the sentence. A doubly attentive decoder distinguishes and balances the inner-word and out-word information utilization in the decode stage. The experiment demonstrates that our approach achieves the SOTA performance.

2 Related work

2.1 Mongolian Morphological Segmentation

Previous works proposed several supervised learning algorithms using artificial features for Mongolian morphological segmentation (Hou et al., 2009; Shi et al., 2015; Ming and Hou, 2011). These approaches usually depend on hand-craft features and cannot handle OOV problems well. Recently, several studies suggested that character-level end-to-end models achieve more superior performance in this task (Narisong et al., 2016; Liu et al., 2018; Zhu, 2018), compared with hand-craft features. Narisong et al. (Narisong et al., 2016) proposed a CRF-based multi-task learning model to deal with Mongolian word segmentation and POS-tagging tasks. Liu et al. (Liu et al., 2018) introduced a two-layers BiLSTM with a limited search strategy and reported new state-of-the-art results for the Mongolian morphological segmentation. These successes reveal that character-level end-to-end neural networks, especially LSTM, can extract and exploit the potential inner-word features. However, these models do not involve the out-word features in the segmentation process. Therefore, this paper proposes an end-to-end model that incorporates the inner-word and the out-word features simultaneously in Mongolian morphological segmentation.

2.2 Self-Attention Network

Self-attention network (SAN), as its name suggests, is a special case of attention mechanism that only needs internal information of a sequence to compute its representation. Thus, it is more flexible at modeling both long-range and local dependencies comparing to RNN/CNN (Yang et al., 2019). SAN has been successfully applied to NLP tasks, including machine translation (Yang et al., 2019; Vaswani et al., 2017; Shen et al., 2018), reading comprehension (Zheng et al., 2018), document summarization (Al-Sabahi et al., 2018), semantic role labeling (Kitaev and Klein, 2018), and constituency parsing (Tan et al., 2017). In this study, we choose the Transformer (Vaswani et al., 2017) as the key architecture of our model. The Transformer consists of two components: an encoder and a decoder. Both encoder and decoder are built by the same layers, multi-head attention, feed-forward, residual connections and normalization sub-layer. In contrast, the decoder contains extra masked multi-head attention comparing to the encoder.

2.3 Integrating Inner-word and Out-word Features

Due to their ability to capture inner-word information of words from the characters, pre-trained character-level static vectors are used in a lot of NLP downstream tasks (Kim et al., 2015; Melamud et al., 2016) which achieve the competitive results with fewer parameters. Other work has also focused on encoding the context around a pivot word dynamically to learn more out-word information (Melamud et al., 2016). Furthermore, it has proved to be helpful when concatenating word-level and character-level knowledge (Devlin et al., 2019; Peters et al., 2018; Peters et al., 2017). In the morphological analysis task of SIGMORPHON 2019, almost all of the researchers use two levels of word representations to capture more inner- and out-word information (McCarthy et al., 2019; Oh et al., 2019; Chaudhary et al., 2019).

3 Approach

This paper proposes a neural network incorporating the inner-word and the out-word features for Mongolian morphological segmentation, as shown in Figure 1. The network consists of two encoders and one decoder. The inner-word encoder uses the self-attention mechanisms to capture the inner-word features of the target word. The out-word encoder employs a two layers BiLSTM network to extract out-word features in the sentence. The decoder adopts a multi-head double attention layer to fuse the inner-word features and out-word features and produces the segmentation result. The following sections describe our approach in detail.

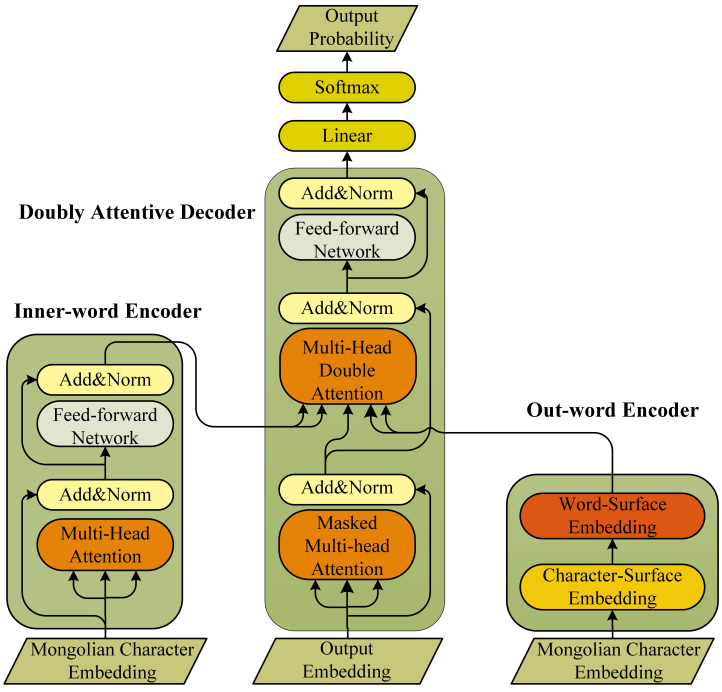

Figure 1: The architecture of the proposed network.

3.1 Inner-word Encoder

3.1.1 Input Embedding

Our model takes a sequence of Mongolian character embeddings $[c_1, c_2, \cdots, c_T]$ as a part of the input, where c_t is from the random initialization character lookup table, which contains a vector for each character. All embeddings are learned jointly with the other parameters of the model. We denote the character position in a word as a vector p_t. Both the character embedding and position embedding have the same dimensionality $\in R^{d_m}$, where the d_m is defined as the model dimension. They are added together at the input layer of our model: $x_t = c_t + p_t$. There are various ways to encode positions. We adopt the signal timing approach from (Kitaev and Klein, 2018) for position embedding p_t, which is formulated as follows:

$$\begin{cases} timing(p, 2i) = sin(\frac{p}{1000^{2i/d_m}}) \\ timing(p, 2i+1) = cos(\frac{p}{1000^{2i/d_m}}) \end{cases} \tag{1}$$

where p represents the character position in a word.

3.1.2 Multi-head Self-attention

The center of this SAN formulation is the multi-head attention sub-layer. The multi-head self-attention sublayer is a variant of dot-product (multiplicative) attention (Luong et al., 2015). Formally, for a single attention head, as illustrated in Figure 2, giving an input matrix X, $X = T \times d_m$, where each row vector x_t corresponds to the t^{th} Mongolian character in the tag word and d_m is the model dimensionality. And the trainable parameter matrices C_Q, C_K, and C_V are used to map an input x_t to three vectors query $q_t = x_t C_Q$, key $k_t = x_t C_K$ and value $v_t = x_t C_V$, where $\{C_Q, C_K, C_V\} \in R^d$ and the d is the number of hidden units of our network. We calculate the probability that character i attending to character j as $p(i \to j) \propto exp(\frac{(q_i \cdot k_j)}{\sqrt{d}})$, and the v_j for all characters that have been attended to are aggregated to form an average value $\overline{v_i}$, $\overline{v_i} = \sum_j p(i \to j) v_j$.

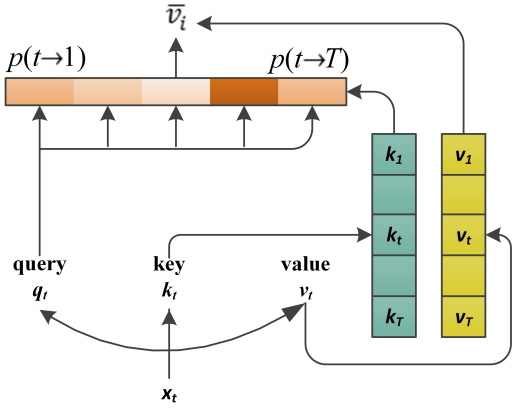

Figure 2: The dot product attention architecture of a single attention head.

The scaled dot-product attention computes the attention scores based on the following mathematical formulation:

$$SingleHead(X) = Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})V \tag{2}$$

where $Q = XC_Q, K = XC_K, V = XC_V$. Finally, all the vectors produced by parallel multi-heads are added together to form a single vector:$M = \sum_{n=1}^{8} SingleHead(X)^n$.This allows a character to gather information from up to 8 remote locations in the sequence at each attentional layer.

3.1.3 Feed-forward network

Our feed-forward sub-layer is simple and follows Vaswani et al. (Vaswani et al., 2017). It consists of two linear layers with hidden ReLU (Rectified Linear Unit) nonlinearity in the middle. Formally, the equation is shown below:

$$FeedForward(X) = W_2 ReLU(W_1 X + b_1) + b_2 \tag{3}$$

where $W_1 \in R^{d_m \times d}$ and $W_2 \in R^{d \times d_m}$ are trainable matrices.

3.2 Out-word Encoder

3.2.1 Character-Surface Embedding

In the out-word encoder, a BiLSTM network is first used to generate the embedding for each word individually. The input is the Mongolian character embedding of each word and the character lookup table is that in section 3.1.1. Let w_i represents the i^{th} word in the sentence, x_t represents the t^{th} character in w_i and e_i^c denotes the character-surface representation. We obtain e_i^c from BiLSTM:

$$e_i^c = \left[\begin{array}{ccc} \overrightarrow{h_T^c} & ; & \overleftarrow{h_1^c} \end{array} \right] \tag{4}$$

where the forward LSTM learns the presentation $\overrightarrow{h_T^c}$:

$$\overrightarrow{h_T^c} = LSTM_{forward}(\overrightarrow{h_{T-1}^c}, x_T) \tag{5}$$

and the backward LSTM learns the presentation $\overleftarrow{h_1^c}$:

$$\overleftarrow{h_1^c} = LSTM_{backward}(\overleftarrow{h_2^c}, x_1) \tag{6}$$

3.2.2 Word-Surface Embedding

In the word-surface embedding layer, we adopt another BiLSTM as the out-word encoder. Its inputs are the word embeddings and the output is the out-word embedding e_i^w of the target word. The e_i^w is shown in Eqs. (7) .

$$e_i^w = \left[\begin{array}{ccc} \overrightarrow{h_i^w} & ; & \overleftarrow{h_i^w} \end{array} \right] \tag{7}$$

where the forward LSTM learns the presentation $\overrightarrow{h_i^w}$:

$$\overrightarrow{h_i^w} = LSTM_{forward}(\overrightarrow{h_{i-1}^w}, e_i^c) \tag{8}$$

and the backward LSTM learns the presentation $\overleftarrow{h_i^w}$:

$$\overleftarrow{h_i^w} = LSTM_{backward}(\overleftarrow{h_{i,2}^w}, e_1^c) \tag{9}$$

3.3 Doubly Attentive Decoder

As illustrated in Figure 1, the core of the doubly attentive decoder (DAD) is the multi-head double attention sublayer and separate feed-forward sublayer. It fuses the inner-word features and out-word features and produces the segmentation result.

3.3.1 Multi-head Double Attention Layer

Doubly attentive decoder integrates two separate attention mechanisms over the inner-word and out-word word features in a single decoder. The main difference between the multi-head doubly attention and the standard multi-head attention focus on the dot-product (multiplicative) attention. Here, the dot product decomposes as $q_i \cdot k_j = q_i^{inner} \cdot k_j^{inner} + q_i^{out} \cdot k_j^{out}$. The detail of the multi-head doubly attention head is shown in Figure 3. can also be viewed as separately applying attention to inner and out, except that the log-probabilities in the two halves are added together prior to value lookup (Kitaev and Klein, 2018).The formalized formula as following:

$$\begin{aligned} &SingleHead(Z^{inner}, Z^{out}) = Attention(Q, K^{inner}, V^{inner}, K^{out}, V^{out}) \\ &= Softmax(\frac{QK^{inner^T}}{\sqrt{d}})V^{inner} + Softmax(\frac{QK^{out^T}}{\sqrt{d}})V^{out} \end{aligned} \tag{10}$$

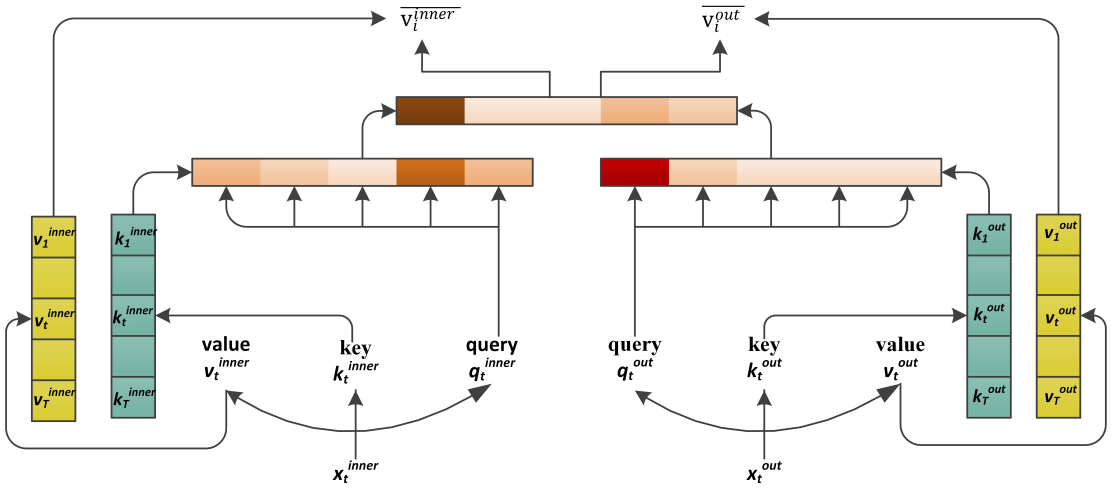

Figure 3: The double dot product attention architecture of a single attention head.

3.3.2 Separate Feed-forward Sublayer

The separate feed-forward sublayer in our doubly attentive decoder is likewise split into two independent portions that operate on inner and out information. It consists of two linear layers with hidden ReLU nonlinearity in the middle and the equation is shown below:

$$FeedForward(x^{inner}) = W_2^{inner} ReLU(W_1^{inner} x^{inner} + b_1^{inner}) + b_2^{inner} \tag{11}$$

$$FeedForward(x^{out}) = W_2^{out} ReLU(W_1^{out} x^{out} + b_1^{out}) + b_2^{out} \tag{12}$$

where $W_1^{inner}, W_1^{out} \in R^{(\frac{d_m}{2} \times d)}$ and $W_2^{inner}, W_2^{out} \in R^{(d \times \frac{d_m}{2})}$ are trainable matrices.

From the perspective of parameters, the trainable parameter matrix of our decoder can be regarded as consisting of two independent matrix blocks:

$$W = \begin{bmatrix} W^{inner} & 0 \\ 0 & W^{out} \end{bmatrix} \tag{13}$$

4 Experiments

4.1 Dataset

Nowadays, the public Mongolian corpus with morphological information are not available. In the experiment, the corpus we used has been annotated by a group of Mongolian native speakers. It includes 20,000 labeled Mongolian sentences, whose length varies from 1 to 84. The average length is 16.63. There are 334,627 words and 24,336 different words in total. The word length is between 1 to 28, and the average length is 8.14. We split it into training dataset (14,000 sentences, 70%), developing dataset (2,000 sentences, 10%) and testing (4,000 sentences, 20%).

We randomly selected 10% of the test set (400 sentences) and named this collection as the review set. Overall, this data set contains 6523 words in total, of which there are 4852 unique words (966 words appear in more than one sentence and three multi-category words).

4.2 Evaluation Metrics

Following the work (Hou et al., 2009; Shi et al., 2015; Ming and Hou, 2011; Liu et al., 2018; Zhu, 2018), we use Precision (P), Recall (R) and $F1$-score to quantitatively evaluate the proposed network. We defined each morpheme as one unit after Mongolian word segmentation. The P is the proportion of the collected units provided by the morphological segmentation model. R is defined as the percentage of corrected units among the reference units.

$$P = \frac{\#(correctly\_tagged\_units)}{\#(model\_tagged\_units)} \times 100\% \tag{14}$$

$$R = \frac{\#(correctly\_tagged\_units)}{\#(manually\_annotated\_units)} \times 100\% \quad (15)$$

$$F1 = \frac{2 \times P \times R}{P + R} \times 100\% \quad (16)$$

4.3 Baseline Approaches

This paper compares our Mongolian morphological segmentation model with the following approaches, including the BiLa* (Zhu, 2018) and the True and pseudo mapping model (Liu et al., 2018).

- BiLa*: BiLa* is an LSTM-based tagging morphological segmentation approach. It applies a BiLSTM layer as the encoder and a LSTM layer as the decoder. We use the same implementation as BiLa* for Mongolian morphological segmentation and keep its default parameters unchanged.

- True and pseudo mapping model: This model is a BiLSTM network for Mongolian morphological segmentation, using a limited search strategy (LSS). We use the same model and hyperparameter as that in (Liu et al., 2018).

4.4 Experiment Setting

The settings of our models are described as follows.

- SAN. The number of our attention layers N is set to 2. The dimension of the model d_m equals 200 and the number of attention heads is set to 8. The dimension sizes of attention query, key and value vectors all are 64. The attention dropout probability is set to 0.2, and the ReLU dropout probability in the feed-forward sublayer is 0.1. The label smoothing technique (Szegedy et al., 2016) is applied with a smoothing value of 0.1 during training.

- BiLSTM. The BiLSTM models have one layer, both for character-surface embedding and word-surface embedding and initialized all of the LSTM's parameters with the uniform distribution between -0.1 and 0.1. The optimization algorithm is stochastic gradient descent without momentum, with a fixed learning rate of 0.8.

5 Results and Discussion

5.1 Comparison with Baselines

Table 2 presents the comparisons of our model and baselines. The proposed model consists of two encoders and a decoder. It achieves 98.35% Precision, 98.18% Recall, and 98.06% $F1$-score, which improves almost 2.56 $F1$-score over the state-of-the-art baseline. This shows that in the Mongolian morphological segmentation model, the use of SAN obtains the better segmentation results by introducing both inner and out features. These results show that a better segmentation result can be obtained by introducing both inner- and out-word features. By using SAN we propose a better Mongolian morphological segmentation model.

Model	P**(%)**	R**(%)**	$F1$**(%)**
BiLa* (Zhu, 2018)	93.93	94.03	93.98
True and pseudo mapping model (Liu et al., 2018)	95.59	94.42	95.50
Ours	98.35	98.18	98.06

Table 2: Comparative results with our model and baseline approaches.

5.2 The Effect of Different Level Features

We evaluated the effectiveness of the different level features in our model and list the results of a single encoder, a standard SAN inner-word encoder or a BiLSTM out-word encoder, as shown in Table 3. The conclusions were obtained as follows:

Model	P(%)	R(%)	$F1$(%)
Transformer (SAN-SAN)	95.83	96.52	96.18
BiLSTM-SAN	97.12	96.59	96.85

Table 3: Performance of different level features.

- Compared with the original Transformer (SAN-SAN) model (Vaswani et al., 2017), the BiLSTM-SAN model obtains better performance (improvements of 1.29, 0.07, and 0.67% in Precision, Recall and $F1$-score, respectively). The results show that our model improves Precision significantly with no reducing Recall. After analysis, we found that the reason for the Precision improvement is the overcutting problem is alleviated. In the experiment, SAN-SAN and BiLSTM-SAN obtain 106 and 87 overcutting words, respectively. The latter has 19 words less than the former. This is because the BiLSTM out-word encoder could capture more contextual features to inhibit overcut.

- In addition, we compare the effects of the Transformer (SAN-SAN) model and the BiLa* model (provided in Table 2). Both of them include an inner-word encoder only and without any decode constraint. The results show that SAN-SAN achieves better performance than BiLa*.Further research shows that the $F1$-score dropped significantly when the word length is longer than 21 in BiLa* because of the bias in LSTM. Although LSTM can solve hard long time lag problems with the gating mechanism (Hochreiter and Schmidhuber, 1997) and the attention mechanism (Luong et al., 2015), the problem of segmentation on long sequence remains. Compared with BiLa*, SANSAN obtains a higher $F1$-score until the word length is longer than 25. This shows that SAN-SAN learns more abundant inner-word information even when dealing with long words, and provide a more flexible way to represent and focus on the crucial information.

5.3 The Effect of Doubly Attentive Decoder

We also evaluate the effect of three decoders (BiLSTM, SAN and DAD) on the same encoder (SAN+BiLSTM), as shown in Table 4. The encoders and decoder are divided by a "-". The conclusions as follows:

Model	P(%)	R(%)	$F1$(%)
SAN+BiLSTM-BiLSTM	95.62	95.44	95.53
SAN+BiLSTM-SAN	97.64	97.05	97.34
SAN+BiLSTM-DAD	98.15	97.98	98.06

Table 4: Performance of different decoders.

- Our model achieves the best performance (98.26% in $F1$-score, and over 2.56% than the state-of-the-art baseline). Compared with the SAN+BiLSTM-BiLSTM and SAN+BiLSTM-SAN models, SAN+BiLSTM-DAD achieves the best effect (over 2.54% and 0.87% in $F1$-score than other two structures). We performed an error analysis of the reviewed set. The result denotes SAN+BiLSTMBiLSTM, SAN+BiLSTM-SAN and SAN+BiLSTM-DAD obtain 121, 84 and 53 overcutting words, respectively. And in the three multi-category words, they segment 1, 1 and 2 words correctly according to the gold standard. Based on the above analysis, the SAN+BiLSTM-DAD model has the best effect on the overcoming overcutting and the multi-category words problem. The gains are due to the explicitly doubly attentive decoder which can distinguish and balance the inner-word and out-word information to find the morpheme boundaries better.

- For the Transformer (SAN-SAN) model, when adding the out-word information (the SAN+BiLSTM-SAN), the high $F1$-score (an the improvement of 1.16% in $F1$-score) is achieved. The main reason is that out-word information can revise some morpheme boundary errors. Compared with the original baseline BiLa* model, the SAN+BiLSTM-BiLSTM model obtains better

performance (improvements of 1.69, 1.41 and 1.55% in Precision, Recall and $F1$-score, respectively). The result has shown once again that no matter which decoder is used, the segmentation performance effectively improves when adding the out-word information.

6 Conclusion

This paper proposed a neural network incorporating inner-word and out-word features for Mongolian morphological segmentation. We employ a standard SAN to encode the inner-word features between the characters in the word and a BiLSTM to encode the implicit out-word features between the target word and other words in the whole sentence. To distinguish and balance utilizing the inner-word and out-word features, we apply a doubly attentive decoder to decode two-level features jointly. The experimental results show that (1) the self-attention mechanism introduced to capture the inner-word information is shown to be more flexible in representing and focusing on the crucial information; (2) the BiLSTM, our out-word information encoder, which has been proved to be sufficient to alleviate the overcutting problem; (3) the doubly attentive decoder is used to distinguish and balance information, allowing the model better to find the morpheme boundaries. Our experiment results show that the proposed model can obtain competitive results compared to early methods. Since the introduction of inner-word and out-word information in the doubly attentive decoder, our method achieves state-of-the-art performance.

Acknowledgements

This work was funded by National Key Research and Development Program of China (Grant No. 2018YFE0122900), National Natural Science Foundation of China (Grant No. 61762069, 61773224, 62066033), Natural Science Foundation of Inner Mongolia Autonomous Region (Grant No. 2019ZD14), and research program of science and technology at Universities of Inner Mongolia Autonomous Region (Grant No. NJZY18237).

References

Kamal Al-Sabahi, Zhang Zuping, and Mohammed Nadher. 2018. A hierarchical structured self-attentive model for extractive document summarization (hssas). *IEEE Access*, 6:24205–24212.

Aditi Chaudhary, Elizabeth Salesky, Gayatri Bhat, David Mortensen, Jaime Carbonell, and Yulia Tsvetkov. 2019. Cmu-01 at the sigmorphon 2019 shared task on crosslinguality and context in morphology. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology (2019)*, pages 57–70.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, volume 1, pages 4171–4186.

Wenting Fan, Hongxu Hou, Hongbin Wang, and Jinting Li. 2017. Improve mongolian-chinese translation by introducing smt information into nmt. In *Proceedings of the International Conference on Computer Science and Application Engineering (CSAE 2017)*, pages 208–217.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hongxu Hou, Qun Liu, and Nasanurtu. 2009. Mongolian word segmentation based on statistical language model. *Pattern Recognition and Artificial Intelligence*, 22(1):108–112.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. 2015. Character-aware neural language models. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI2015)*, volume arXiv:1508.066156. version 5.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL2018)*, volume arXiv:1805.01052.

R. Kullmann and D. Tserenpil. 2008. *Mongolian Grammar*. ADMON Co.,Ltd, Mongolia.

Na Liu, Junyi Wang, and Guiping Liu. 2012. Query expansion based on mongolian semantics. In *Proceedings of the Third World Congress on Software Engineering IEEE Computer Society*, pages 25–28.

Rui Liu, Feilong Bao, Guanglai Gao, and Yonghe Wang. 2017. Mongolian text-to-speech system based on deep neural network. In *Proceedings of the Man-Machine Speech Communication*, volume 807, pages 99–108.

Na Liu, Xiangdong Su, Guanglai Gao, and Feilong Bao. 2018. Mongolian word segmentation based on three character level seq2seq models. In *Proceedings of the Neural Information Processing*, volume 11305.

Minh-Thang Luong, Hieu Pham, and Christoper Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Arya McCarthy, Katerina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The sigmorphon 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology (2019)*, volume arXiv:1910.11493, pages 229–244.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*, pages 51–61.

Yu Ming and Hongxu Hou. 2011. *Researching of Mongolian word segmentation system based on dictionary, rules and Language model*. M.S.Thesis, Inner Mongolia University, Hohhot, Inner Mongolia, China.

Narisong, Hu Qin, and Qiliger. 2016. Research on crf-based mongolian word segmentation and pos-tagging. *Journal of Inner Mongolia University (Philosophy and Social Sciences)*, 48(2):23–28.

Byung-Doh Oh, Pranav Maneriker, and Nanjiang Jiang. 2019. Thomas: The hegemonic osu morphological analyzer using seq2seq. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology (2019)*, pages 80–86.

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, volume arXiv:1802.05365, pages 4171–4186. version 2.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2018. Bi-directional block self-attention for fast and memory-efficient sequence modeling. In *Proceedings of the International Conference on Learning Representations (ICLR2018)*, volume arXiv:1804.00857.

Jianguo Shi, Hongxu Hou, and Feilong Bao. 2015. Research on slavic mongolian word segmentation based on dictionary and rule. *Journal of Chinese Information Processing*, 29(1):197–202.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and ZB Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2017. Deep semantic role labeling with self-attention. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI2017)*, volume arXiv1712.01586.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, volume arXiv:1706.03762.

Weihua Wang, Feilong Bao, and Guanglai Gao. 2016. Mongolian named entity recognition with bidirectional recurrent neural networks. In *Proceedings of the IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 495–500, Los Alamitos, CA, USA.

Weihua Wang, Feilong Bao, and Guanglai Gao. 2019. Learning morpheme representation for mongolian named entity recognition. *Neural Processing Letters*, 50.

Zhenxin Yang, Miao Li, Lei Chen, and Kai Sun. 2016. A morpheme-based weighting for chinese-mongolian statistical machine translation. In *Proceedings of the IEICE Transactions on Information and Systems*, pages 2843–2846.

Baosong Yang, Jian Li, Derek Wong, Lidia Chao, Xing Wang, and Zhaopeng Tu. 2019. Context-aware self-attention networks. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI 2019)*, volume 33, pages 387–394.

Yukun Zheng, Dan Li, Zhen Fan, Yiqun Liu, Min Zhang, and Shaoping Ma. 2018. T-reader: A multi-task deep reading comprehension model with self-attention mechanism. *Journal of Chinese Information Processing*, 32(11):128–134.

Shunle Zhu. 2018. A neural attention based model for morphological segmentation. *Wireless Personal Communications*, 102(4):2527–2534.