Improving sentence simplification model with ordered neurons network

Chunhui Deng | Lemin Zhang | Huifang Deng

1School of Computer Engineering, Guangzhou College of South China University of Technology, Guangzhou, China
2School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

Abstract
Sentence simplification is an essential task in natural language processing and aims to simplify complex sentences while retaining their primary meanings. To date, the main research works on sentence simplification models have been based on sequence-to-sequence (Seq2Seq) models. However, these Seq2Seq models are incapable of analysing the hierarchical structure of sentences, which is of great significance for sentence simplification. The problem can be addressed with an ON-MULTI-STAGE model constructed based on the improved MULTI-STAGE encoder model. In this model, an ordered neurons network is introduced and can provide sentence-level structural information for the encoder and decoder. A weak attention connection method is then employed to make the decoder use the sentence-level structural details. Experimental results on two open data sets demonstrated that the constructed model outperforms the state-of-the-art baseline models in sentence simplification.

1 | INTRODUCTION
Sentence simplification is an essential text-generation task in the field of natural language processing (NLP). It aims to simplify the linguistic complexity of the source sentence while retaining the main idea of the sentence and has many applications in practice. For example, it can help people with low-literacy skills acquire information effectively [1, 2] and improve the performance of other NLP tasks, such as text summarization [3] etc.

Traditional sentence simplification models focus primarily on paraphrasing words [4–6], deleting unimportant phrases/words [7–9], or dividing a long sentence into shorter sentences [2, 10]. But such models cannot be trained end-to-end and depend too much on artificial rules. Inspired by the great success of sequence-to-sequence (Seq2Seq) models in machine translation tasks [11, 12], recent works [13–15] have constructed similar Seq2Seq sentence simplification models along with attention mechanisms. Zhang et al. [13] further improved the Seq2Seq model with reinforcement-based policy gradient approaches. Vu et al. [14] improved the encoder architecture of the Seq2Seq model with augmented memory capacities called neural semantic encoders (NSEs). Zhang and Deng [15] modified the encoder with a multi-stage encoder to further improve the Seq2Seq model. However, these models still have several deficiencies: (1) Their network structures [13–15] cannot make use of the neuron-ordered information; (2) Their encoder and decoder [13–15] fail in considering the hierarchical structural information of the sentence, which is of great significance in sentence simplification.

To address the above problems and follow the work of [15], we have proposed an ON-MULTI-STAGE model based on the improved MULTI-STAGE encoder model [15] by introducing an ordered neurons (ON) network [16]. The proposed model goes beyond conventional long short-term memory (LSTM)/gated recurrent units (GRUs) by introducing the ON network structure to optimize the encoder of the MULTI-STAGE model. The ON network can express the hierarchical structural information of sentence with the neuron-ordered information. This network architecture has been shown to be effective in language modelling task [16].

The encoder of the proposed model works in three stages: N-gram reading, glance-over, and final encoding stages. The
N-gram reading stage extracts the N-gram grammatically context-related convolutional word vector. The glance-over stage uses the ON network to catch the hierarchical structural information of the sentence. The final encoding stage takes advantage of the convolutional word vector and the hierarchical structural information of the sentence to encode the source sentence more effectively. In addition, the proposed model employs a weak attention connection mechanism to facilitate its decoder to use the hierarchical structural information of the sentence. To the best of our knowledge, the ON-MULTI-STAGE model is the first Seq2Seq model to use ON networks and the first model to use the ON network for sentence simplification.

The proposed model is first evaluated on two open data sets, Newsela [17] and WikiLarge [13]. The experimental results show that the proposed ON-MULTI-STAGE model is significantly better than other benchmark models. Then the explorative experiments are conducted inside the model. Finally, the comparison is made on the outputs of the proposed model with the MULTI-STAGE model [15]. The comparison also shows that the proposed model is more effective on sentence simplification.

The rest of this paper is organized as follows: Section 2 provides a review of related works. Section 3 presents the proposed ON-MULTI-STAGE model. Section 4 shows experimental results and comparisons with benchmark models. The last section concludes this paper.

2 RELATED WORKS

In the past, traditional sentence simplification models focused mainly on simplifying syntactically complex structures, substituting complex words with simpler ones, or deleting unimportant parts of source sentences. These models usually focus solely on individual aspects of the sentence simplification problem. Some models conduct only syntactic simplification [4, 18], while others deal with lexical simplification solely by replacing rare words with simpler WordNet synonyms [5, 6].

Recently, some works have regarded the sentence simplification task as a monolingual text-to-text generation task. These works use models based on statistical machine translation to solve for sentence simplification. For example, Woodsend et al. [19] used the quasi-synchronous grammar framework to formulate sentence simplification and turned to linear programming to score candidate simplifications. Wubben et al. [20] proposed a two-stage model. In the first stage, they trained a phrase-based machine translation model with complex–simple sentence pairs. In the second stage, they reranked the topK outputs of the machine translation model according to the dissimilarity of outputs to the source sentence. Following the work of [20], Narayan and Gardent [21] proposed a hybrid model. In this model, a Boxer [22] based probabilistic module first does sentence segmentation and deletion operations over discourse representation structures. Next, the output sentences are further simplified by the model similar to [20].

With the development of deep learning and neural network, neural Seq2Seq models have been successfully applied in many Seq2Seq generation works, such as machine translation [23, 24] and summarization [25]. Inspired by the success of Seq2Seq models in these NLP applications, Zhang et al. [13] proposed a deep reinforcement learning sentence simplification model (DRESS) that integrates the attention-based Seq2Seq model with reinforcement learning for the reward of simpler outputs. They then further improved the model with lexical simplification and proposed the DRESS–LS model. Vu et al. [14] proposed to use a memory-augmented recurrent neural network architecture, called NSNs [26] to improve the encoder of conventional LSTM- or GRU-based Seq2Seq models. Zhang and Deng [15] went beyond the conventional single-stage encoder-based Seq2Seq models and proposed a multi-stage encoder Seq2Seq model (MULTI-STAGE). In addition, based on a pointer-generator [27] Seq2Seq model, Guo et al. [28] further improved its text entailment and paraphrasing capabilities with multi-task learning. But this multi-task learning method heavily relies on data sets other than sentence simplification tasks. Nishihara et al. [29] proposed a controllable sentence simplification model incorporating a lexical constraint loss into the Seq2Seq model. But their model depends heavily on data set constituents because the model needs the sentence-level label of the target sentence. These Seq2Seq models have achieved a certain degree of good results in the sentence simplification task.

However, the network structures of these conventional Seq2Seq models cannot take advantage of neuron-ordered information. And the encoder and decoder of these models do not consider the hierarchical structural information of the sentence that is of great significance for sentence simplification. To solve these problems, we have proposed an ON-MULTI-STAGE model.

3 METHODOLOGY

An ON-MULTI-STAGE model is proposed. The proposed model goes beyond the conventional neural network structures and uses an ON network [16] structure to optimize the encoder of the improved MULTI-STAGE model [15]. To the best of our knowledge, the ON-MULTI-STAGE model is the first Seq2Seq model that uses an ON network and the first model that uses the ON network for sentence simplification. In this section, we first introduce the concept of an ON network, and then present the ON-MULTI-STAGE model.

3.1 Ordered neurons

For conventional neural networks, such as LSTM, their neurons are usually disordered. Therefore, these networks cannot express the hierarchical structural information of sentence with
the neuron-ordered information. Unlike these neural networks, the ordered neurons long short-term memory network (ONLSTM) [16] can extract the hierarchical structural information of sentences by using the ordered information of neurons. Given an example sentence like 'Jay comes from China', the hierarchical structure tree of this sentence is shown in Figure 1.

In the ON network, the neuron-ordered information Figure 2 is used to provide the hierarchical structure of the sentence (Figure 1). As shown in Figure 2, when processing the sentence, the information contained in different dimensions of the ONs should correspond to the hierarchical structure tree of the sentence. For example, the low-dimensional neurons (e.g., the word-level neurons in Figure 2) mainly express word-level information (e.g., the leaf node of the tree in Figure 1), and the higher dimensional neurons (e.g., the phrase-level or sentence-level neurons in Figure 2) mainly represent the phrase or root sentence-level information (e.g., the phrase-level node or root of the tree in Figure 1). This is precisely because the ON network can provide the hierarchical structural details of the sentence through the ordered information of neurons; the neurons of the ON network are ordered, which is different from the neurons of conventional neural networks.

3.2 | ON-MULTI-STAGE model

Sentence simplification can be viewed as a sequence-to-sequence text generation task, where the target sentence is much simpler than the source sentence. The MULTI-STAGE model [15] modified the encoder of the conventional Seq2Seq model by using a multi-stage encoder and has shown its effectiveness in sentence simplification. However, this model has not considered the hierarchical structure information of the sentence, which is of great significance for sentence simplification. Because of this, a further improvement on the MULTI-STAGE model attempts to integrate the ON network into the encoder of the MULTI-STAGE model. We name this model the ON-MULTI-STAGE model.

Figure 3 shows the overall structure of the proposed model. Given a complex source sentence $X = (x_0, x_1, \ldots, x_i, \ldots, x_{|L|})$, where $x_i$ denotes $i$-th word-embedding vector in the source sentence of length $|L|$, the proposed model learns to generate its simplified target $Y = (y_1, y_2, \ldots, y_j, \ldots, y_{|l|})$, where $y_j$ denotes the $j$-th word in the target sentence of length $|l|$. As shown in Figure 3, the encoder works in three stages. The first stage is the N-gram reading stage, where N represents the size of the convolution kernel. In this stage, a convolution operation on the word-embedding vector of the source sentence is used to obtain the convolution word-embedding vector. The second stage is the glance-over stage, which is built from the ONLSTM network. In this stage, the ONLSTM network (the green -coloured part in Figure 3) extracts the hierarchical structural information of the sentence for the final stage of the encoder of the proposed model. The third stage is the final encoding stage. In this stage, the proposed model extracts the final encoding of the source sentence based on the information obtained in the first two stages.

3.2.1 | Encoder of ON-MULTI-STAGE model

The first stage of the encoder is the N-gram reading stage. In this stage, the proposed model mainly obtains the convolutional word-embedding vector of the source sentence by the convolution operation. Compared with the conventional word-embedding vector, the convolutional word-embedding vector has N-gram contextual relevance (blue coloured part in
The convolutional word-embedding vector matrix $\textbf{EX}$ can be calculated as follows:

$$\textbf{EX} = (e_{x_1}, e_{x_2}, \ldots, e_{x_t}, \ldots, e_{x_{|L|}})$$  \hspace{1cm} (1)

$$e_{x_t} = W_{mg} \left[ \begin{array}{c} x_{t-[N/2]} \\ x_{t-[N/2]+1} \\ \vdots \\ x_{t+[N/2]} \end{array} \right] + b_{ng}$$  \hspace{1cm} (2)

where $W_{mg}$, $b_{ng}$ are trainable parameters of the convolution kernel, $N$ is the convolutional kernel size, $e_{x_t}$ is the $t$-th convolutional word-embedding vector of the $\textbf{EX}$, and $x_t$ is the $t$-th word-embedding vector of the source sentence.

The second stage is the glance-over stage. In this stage, the proposed model uses ONLSTM [16] (the green-coloured part in Figure 3) to extract the hierarchical structural information of the source sentence based on information obtained in the N-gram reading stage and the source input. Similar to LSTM, ONLSTM also contains three gate structures. Specifically, these gate structures of $t$-th time step can be calculated as follows:

$$f_{ot} = \sigma[W_{of}(x_t, e_{x_t}) + U_{of}s_{t-1} + b_{of}]$$  \hspace{1cm} (3)

$$i_{ot} = \sigma[W_{oi}(x_t, e_{x_t}) + U_{oi}s_{t-1} + b_{oi}]$$  \hspace{1cm} (4)

$$o_{ot} = \sigma[W_{oo}(x_t, e_{x_t}) + U_{oo}s_{t-1} + b_{oo}]$$  \hspace{1cm} (5)

where $W_{of}$, $W_{oi}$, $W_{oo}$, $U_{of}$, $U_{oi}$, $U_{oo}$, $b_{of}$, $b_{oi}$, and $b_{oo}$ are all trainable parameters; $\sigma$ is sigmoid function; and $s_{t-1}$ is the hidden state of the encoder’s glance-over stage at $(t-1)$th time step. The word-level information vector $\hat{c}_t$ of the source sentence at the $t$-th time step is calculated as follows:

$$\hat{c}_t = \tanh[W_{oc}(x_t, e_{x_t}) + U_{oc}s_{t-1} + b_{oc}]$$  \hspace{1cm} (6)

where $W_{oc}$, $U_{oc}$, and $b_{oc}$ are all trainable parameters. Before calculating the neurons $c_t$ of ONLSTM at the $t$-th time step, we first define two level constants, $\text{index}_x$ and $\text{index}^c_t$. $\text{index}_x$ represents the level of the word processed at the $t$-th time step (or current time step), whereas $\text{index}^c_t$ represents the level of historical information of the sentence at the $t$-th time step.

When $\text{index}_x \geq \text{index}^c_t$ there is an intersection of the processed word and the historical information of the sentence. Thus, the processed word information should be integrated into the intersection part to update the phrase-level information. At this time, the update method of the neurons $c_t$ is

$$c_t = \begin{pmatrix} c_{t-1, [\text{index}, \text{dim}]} \\ f_{ot, [\text{index}, \text{index}_x]} \times c_{t-1, [\text{index}, \text{index}_x]} + i_{ot, [\text{index}, \text{index}_x]} \times \hat{c}_t, [\text{index}, \text{index}_x] \\ \hat{c}_t, [0, \text{index}_x] \end{pmatrix}_0$$  \hspace{1cm} (7)

The dim in Equation (7) is the dimension of $c_t$. The dimension of the neurons in brackets gradually increases from bottom to top. And the smaller the dimension, the lower the level of information recorded by the neurons. (7) shows that, the information of low-level neurons (e.g., word-level neurons in Figure 2) is updated with the information of the word-level (e.g., $\hat{c}_t$) at the current time step. The information of high-level neurons (e.g., sentence-level neurons in Figure 2) is updated with the sentence-level information (e.g., $c_{t-1}$), and the intersection part is updated with the integration of the word-level information with the sentence-level information.

When $\text{index}_x < \text{index}^c_t$ there is no intersection of the processed word and the historical information of the sentence. Therefore, the intersection does not need to be written, but the original initial state (zeros state) is kept. At this time, the update method of the neuron $c_t$ is

$$c_t = \begin{pmatrix} c_{t-1, [\text{index}, \text{dim}]} \\ 0, [\text{index}, \text{index}_x] \\ \hat{c}_t, [0, \text{index}_x] \end{pmatrix}_0$$  \hspace{1cm} (8)

As in (7), the dimension of the neurons in brackets gradually increases from bottom to top in Equation (8). And the smaller the dimension number is, the lower the level of information recorded by the neurons. However, unlike (7), the intersection part in Equation (8) maintains the original state.

Now let us seek the solution of $c_t$. First, we define $1_{\text{index}}$ as the one-hot vector whose $\text{index}$-th element is 1, and let $\text{index}_x$
correspond to $1_{\text{indexy}}$ and $1_{\text{indexx}}$; then neuron $c_i$ can be calculated as follows:

$$
\tilde{f} = \text{cumsum}(1_{\text{indexy}}) 
$$

$$
\hat{i} = 1 - \text{cumsum}(1_{\text{indexx}}) 
$$

$$
\omega_{\text{ot}} = \tilde{f}_{\text{ot}} \times \hat{i}_{\text{ot}} 
$$

$$
c_i = \left( \frac{\hat{i} - \omega_{\text{ot}}}{\text{low-level}} \times \hat{c}_i + \omega_{\text{ot}} \times \left( \frac{\tilde{f}_{\text{ot}} \times c_{t-1} + i_{\text{ot}} \times \hat{c}_t}{\text{intersection part}} \right) \right) + \left( \frac{\tilde{f} - \omega_{\text{ot}}}{\text{high-level}} \times c_{t-1} \right) 
$$

$$
\text{cumsum}(\{z_1, z_2, \ldots, z_n\}) = [z_1, z_1 + z_2, \ldots, z_1 + z_2 + \ldots + z_n] 
$$

where $z$ represents the independent variable of the cumsum function. The value of the master forget gate $\tilde{f}$ is monotonically increasing from 0 to 1, and the value of the master input gate $\hat{i}$ is monotonically decreasing from 1 to 0. The two master gates serve as high-level control for the update operations of $c_i$.

In Equations (11) and (12), $\omega_{\text{ot}}$ represents the intersection part, $\hat{i} - \omega_{\text{ot}}$ represents the low-level part, and $\tilde{f} - \omega_{\text{ot}}$ represents the high-level part. Thus, Equation (12) can represent the combined result of Equations (7) and (8).

However, $1_{\text{indexy}}$ and $1_{\text{indexx}}$ are discrete variables, and it is not trivial [30] to compute gradients when a discrete variable occurs in the computation graph. Hence, in practice, $1_{\text{indexy}}$ and $1_{\text{indexx}}$ can be obtained by the softmax function (also known as the normalized exponential function, which is an extension of the binary classification function of sigmoid to multiple classifications):

$$
1_{\text{indexy}} = \text{softmax} \left[ W_{\tilde{f}}(x_i, e_{xi}) + U_{\tilde{f}}s_{t-1} + b_{\tilde{f}} \right] 
$$

$$
1_{\text{indexx}} = \text{softmax} \left[ W_i(x_i, e_{xi}) + U_is_{t-1} + b_i \right] 
$$

where $W_{\tilde{f}}, U_{\tilde{f}}, W_i, U_i, b_{\tilde{f}}$ and $b_i$ are all trainable parameters. After solving $c_i$, the $t$-th time step hidden state of the encoder’s glance-over stage $s_i$ can be calculated as

$$
s_i = o_{\text{ot}} \tanh(c_i) 
$$

The last stage of the encoder is the final encoding stage. In this stage, the proposed model uses a bidirectional LSTM to transform all input into the ultimate final representation states $\mathbf{H}_t = (b_{1t}^F, b_{2t}^F, \ldots, b_{|L|}^F)$, which can be calculated as follows:

$$
\overrightarrow{b}_t \equiv \text{LSTM} \left[ (x_t, e_{xi}, s_i, s_{|L|}), \overrightarrow{b}_{t-1} \right] 
$$

$$
\overleftarrow{b}_t \equiv \text{LSTM} \left[ (x_t, e_{xi}, s_i, s_{|L|}), \overleftarrow{b}_{t-1} \right] 
$$

$$
\overrightarrow{b}_t = \left[ \overrightarrow{b}_t, \overleftarrow{b}_t \right] 
$$

where $\overrightarrow{b}_t$ is the $t$-th time step forward propagation hidden state, and $\overleftarrow{b}_t$ is the $t$-th time step backwards propagation hidden state. We concatenate them as the $t$-th time step final representation state $b_t^F$ of the final encoding stage. The superscript $F$ denotes hidden states of the final encoding stage’s encoder.

### 3.2.2 Attention mechanism

For the model to generate attention distribution, the hierarchical structural information of the sentence can also be considered. A weak attention connection method is applied in the glance-over stage. The attention weights $\alpha_{ti}$ of the proposed model can be calculated as

$$
\alpha_{ti} = \frac{\text{score}(b_{ti}^F, s_i, d_i)}{\sum_j \text{score}(b_{tj}^F, s_i, d_i)} 
$$

$$
\text{score}(b_{ti}^F, s_i, d_i) = \exp \left[ \nu^T \tanh(\beta_1 W_{out} s_i^0 b_{ti}^F + \beta_2 W_{out} s_i + W_{out} b_{ti}^D) \right] 
$$

where $\nu^T, W_{out}, W_{out}^2$, and $W_{out}^3$ are all trainable parameters. $\beta_1, \beta_2$ are two hyperparameters and $\beta_1 + \beta_2 = 1$, where $\beta_1$ is close to the upper bound of 1, $\beta_2$ is close to the lower bound of 0. As $\beta_2$ is the attention connection parameter corresponding to the glance-over stage and close to the lower bound of 0, this attention connection is weak. It is worth noting that $b_{ti}^F$ is the $t$-th vector of the $\mathbf{H}_t$, and $d_i$ is the $t$-th time step hidden state of the decoder. The network structure of the decoder is the LSTM. Thus, $d_i$ can be calculated as follows:

$$
d_i = \text{LSTM}(y_{t-1}, d_{t-1}) 
$$
3.2.3 | Output of model

To allow the proposed model to generate important words (such as person name or place name) not in the vocabulary but in the source sentence, we incorporate the model with a pointer-copy mechanism [27]. The proposed model then generates the t-th time step final output $P_{\text{final}}(y_t)$ by combining the output of the decoder $P_{\text{decoder}}(y_t)$ and the attention distribution $P_{\text{att}}(y_t)$:

$$P_{\text{final}}(y_t) = p_{\text{gen}}P_{\text{decoder}}(y_t) + \left(1 - p_{\text{gen}}\right)P_{\text{att}}(y_t)$$

where $p_{\text{gen}}$ is the pointer-copy probability and can be calculated as

$$p_{\text{gen}} = \sigma(W_{\text{gen}}c_t + U_{\text{gen}}d_t + V_{\text{gen}}y_{t-1} + b_{\text{gen}})$$

where $W_{\text{gen}}, U_{\text{gen}}, V_{\text{gen}},$ and $b_{\text{gen}}$ are all trainable parameters. The output $P_{\text{decoder}}(y_t)$ of the decoder in (24) is calculated as

$$P_{\text{decoder}}(y_t) = \text{softmax}[W_{\text{decode}}\tanh(U_{\text{decode}}d_t + V_{\text{decode}}c_t)]$$

where $W_{\text{decode}}, U_{\text{decode}},$ and $V_{\text{decode}}$ are all trainable parameters. The attention distribution $P_{\text{att}}(y_t)$ in (24) can be obtained by (20).

The objective function $L(\theta)$ of the proposed model is the negative log-likelihood:

$$L(\theta) = -\frac{1}{K} \sum_{bs=1}^{K} \sum_{i=1}^{|I|} \log P_{\text{final}}(y_{t|bs}^{(i)}, y_{<t|bs}^{(i)}, X^{(bs)}, \{\theta\})$$

where $K$ is the batch size, $bs$ denotes the $bs$-th batch, and $X$ is the source sentence; $\{\theta\}$ is the set of all trainable parameters determined through the training and is not required for presetting.

4 | EVALUATION SETUP

4.1 | Data sets

Two benchmark simplification data sets of Newsela and WikiLarge are used for evaluating the proposed model. Newsela is an artificial data set proposed by Xu et al. [17]. This data set is constructed by professional news editors based on 1130 news articles and written at four different reading levels (level 4 is the most simplified, and level 0 is the most complex) to meet children’s reading standards in different grades. Following Zhang and Lapata’s approach [13], we removed adjacent level sentence pairs (such as 0–1, one to two, and 2–3), which are too similar to each other. After this filtering, we are left with 94,208 sentence pairs for training, 1129 for validation, and 1077 for the test; WikiLarge is a non-artificial data set constructed by Zhang and Lapata [13]. It is a large English Wikipedia corpus. It consists of 296,402 complex-simple sentence pairs for training. As for validation and test, following Zhang and Lapata, we use the validation and test sets created in the work of Xu et al. [23].

4.2 | Evaluation metrics

Following previous works [13–15], the proposed model is evaluated against the standard evaluation metrics system output against references and against the input sentence (SARI) [23]. The SARI metrics compare the model’s output with both the reference and the input sentence.

4.3 | Benchmark models

The comparisons are made with the HYBRID model proposed by Narayan et al. [21], DRESS and DRESS-LS models by Zhang et al. [13], NESLSTM model by Vu et al. [14], and MULTI-STAGE model by Zhang et al. [15]. Among them, the HYBRID is a non-neural network benchmark model, and the others are neural network benchmark models.

4.4 | Training details

The proposed model is implemented based on OpenNMT [31], an open-source toolkit, and all trainable parameters are uniformly initialized within the range of [-0.1, 0.1]. Adagrad [32] algorithm is used to optimize the proposed model with learning rate 0.15 and an initial accumulator value of 0.1. Batch size is 64. In order to avoid gradient explosion, a maximum gradient norm of two is applied. The best performing parameters of our model are set or found as follows: both glance-over’s encoder and decoder have 512 hidden neurons; the final encoding stage’s encoder has 256 hidden neurons; The window size N of the N-gram reading stage is three; hyperparameter $\beta_1$ and $\beta_2$ are tuned on validation set and found that the best value for $\beta_1$ is 0.9 and for $\beta_2$ is 0.1. To limit vocabulary size, we included only the 50,000 most frequent words in the vocabulary and the other words were replaced with <UNK> token which is proposed by Jean et al. [33].
5 | EXPERIMENTS

In this section, we first compared the proposed model with benchmark models. And then, we conducted exploration experiments inside the model. Finally, we compared the outputs of the proposed model with the best of the relevant benchmark models. Table 1 shows the experiment results. Because Newsela contains high-quality simplifications created by professional editors, the Newsela result is discussed first.

5.1 | Experimental model comparisons

In this experiment, the proposed model is compared with other benchmark models on Newsela and WikiLarge. As shown in Table 1, the proposed model achieved the highest SARI of 31.49, which is better than all benchmark models, and outperforms the best neural model (MULTI-STAGE) by 1.28 in SARI score, an increase (improvement) of 4.24%. It also outperforms the best non-neural model (HYBRID) by 1.49, an increase of 4.97%.

On the non-artificial WikiLarge data set, the proposed model achieves its highest SARI, 38.22. Specifically, it outperforms the best neural model (MULTI-STAGE) by 0.1 in SARI score, an increase of 0.26%, and outperforms the best non-neural model (HYBRID) by 6.82, a significant increase of 21.72%. This indicates that the model that considers the hierarchical information of the sentence is effective and performs well in sentence simplification tasks.

5.2 | Exploring experiments within the proposed model

In this subsection, we conduct exploration experiments within the proposed model because WikiLarge comprises automatically aligned sentence pairs where errors不可避免ly exist, and the uniform writing style of WikiLarge data set may cause models to generalize poorly [17]. In contrast, Newsela contains high-quality simplifications created by professional editors. Thus, to conduct the exploration experiments more efficiently, the experiments are mainly carried out on Newsela.

5.2.1 | Effect of window size in N-gram reading stage

To investigate the effect of the window size N in the N-gram reading stage, we tried different settings of N as \{1, 2, 3, 4, 5\} and conducted the experiments on the Newsela. The effect of window size N on the simplification task is measured by SARI. The experimental results are shown in Figure 4 and clearly show that \(N = 3\) yields the best performance. \(N = 4\) can reach comparable performance, but the larger N has lower performance than that of \(N = 3\), and all do better than \(N = 1\). This result implies that it is better for the proposed model to represent a word by using the other words around it, and it can help the glance-over stage to better extract the hierarchical structural information of sentences.

5.2.2 | Effect of weak attention

To evaluate the impact of the scale parameters \(\beta_1\) and \(\beta_2\) in Equation (21), we explored different settings of \{(\(\beta_1 = 1\), \(\beta_2 = 0\)), (\(\beta_1 = 0.95\), \(\beta_2 = 0.05\)), (\(\beta_1 = 0.9\), \(\beta_2 = 0.1\)), (\(\beta_1 = 0.8\), \(\beta_2 = 0.2\)), (\(\beta_1 = 0.7\), \(\beta_2 = 0.3\))\} and conducted the experiment on the Newsela. SARI measures the effect of the scale parameters on the simplification task. The experiment result is shown in Figure 5. It clearly shows that the set of (\(\beta_1 = 0.95\), \(\beta_2 = 0.05\)) performs best. It suggests that (1) scale parameters \(\beta_1\) and \(\beta_2\) are better to set around 0.95:0.05; (2) the weak

### Table 1 Comparison of output examples on test set

| Model                     | Newsela [17] | WikiLarge [13] |
|---------------------------|--------------|-----------------|
| Non-neural network benchmark model |               |                 |
| HYBRID                    | A            | 30.00           | 31.40           |
| Neural network benchmark models |               |                 |
| DRESS                     | B            | 27.37           | 37.08           |
| DRESS-LS                  | C            | 26.63           | 37.27           |
| NESLSTM                   | D            | 29.58           | 36.88           |
| MULTI-STAGE               | E            | 30.21           | 38.12           |
| Our model                 | F            | 31.49           | 38.22           |
| Our model Improvement (%) compared with non-neural network benchmark models | \(\beta A\) | 4.97           | 21.72           |
| Improvement (%) compared with other neural network models of the best | \(F = \max\{B,C,D,F\}\) | 4.24           | 0.26             |
connection approach can help the decoder of the proposed model using the hierarchical structural information of the sentence, and therefore can improve the model performance to some extent.

5.2.3 | Dimension of ordered Neurons

To analyse the impact of the dimension of ONs in the glance-over stage on sentence simplification task. We explored different dimension setting sets of dim {64, 128, 192, 256, 384, 512} and conducted the experiment on the Newsela. The results are also measured by SARI and shown in Figure 6. The SARI score rises with the increase in the dimension of ONs. Meanwhile, when the dimension is higher than 384, the improvement trend becomes slow. It means that (1) The increase in the dimension dim of ONs can improve the effect of model sentence simplification tasks; (2) If there is a need to save memory space as much as possible without reducing the effect of sentence simplification tasks too much, the dimension dim of ONs can be set between 256 and 384, and the relatively good results can be achieved as well.

5.3 | Model output example analyses

Because the proposed model is improved based on the MULTI-STAGE model, to demonstrate the improvement effect of the model more intuitively, this section compares several output examples of the two models on the test set in Table 2.

As shown in Table 2, in the first example, the ON-MULTI-STAGE model can replace the word ‘assist’ with ‘help’. In the second example, the ON-MULTI-STAGE model can further delete unnecessary words in the sentence. In the last example, the ON-MULTI-STAGE model can use the simple word ‘study’ to replace most of the content of the input sentence while the original meaning is maintained. As can be seen from these examples, the ON-MULTI-STAGE model improved the MULTI-STAGE model further. However, in the first example, it can be seen that the model still has some inadequacy in maintaining sentence grammatical accuracy. For example, after replacing the word ‘assist’ with ‘help’, the word ‘an’ is not replaced with ‘a’.

6 | CONCLUSIONS

We proposed an ON-MULTI-STAGE model based on the MULTI-STAGE model. Differing from the conventional sequence-to-sequence model, the ON-MULTI-STAGE model can catch the sentence hierarchical structural information through the neuron-ordered information. To the best of our knowledge, the ON-MULTI-STAGE model is the first Seq2Seq model to introduce the ON network and the first model to use the ON network for sentence simplification. To verify the effectiveness of the proposed model, we evaluated it on two benchmark simplification data sets. The results of the experiments showed that the proposed model outperforms all benchmark models and can significantly reduce the complexity of the input sentence while preserving the meaning of the source sentence. Future work may need to explore the following aspects: (1) consideration of using multi-task learning method to improve the grammatical accuracy of the model in generating target sentences; (2) improvement in the decoder’s generating capability by utilizing multi-stage decoding.
### Table 2
Comparison of output examples on test set

| INPUT (Example 1) | Colombia is getting an assist from the US government, which is providing training and expertise to help it clear regulatory hurdles. |
|-------------------|----------------------------------------------------------------------------------|
| MULTI-STAGE       | Colombia is getting help from the US government.                                  |
| ON-MULTI-STAGE(Ours) | In 2013 alone, Chinese developers spent nearly $24 billion on theme park construction, according to AECOM, an engineering company in California. |
| INPUT(example 2)   | In 2013 alone, Chinese developers spent nearly $24 billion on theme park construction, according to AECOM. |
| MULTI-STAGE       | They could not afford to pay for their son’s roughly $10,000 tuition at the University of Texas. |
| ON-MULTI-STAGE(Ours) | They could not afford to pay for their son’s study. |

## ACKNOWLEDGEMENTS
This work was supported in part by Department of Education of Guangdong Province under Special Innovation Program (Natural Science), grant number 2015KTSCX183 and in part by South China University of Technology under ‘Development Fund’ with fund number x2js-F8150310.

## REFERENCES
1. Watanabe, W.M., et al.: Facilita: reading assistance for low-literacy readers. International Conference on Design of Communication, 29–36 (2009)
2. Siddharthan, A.: Syntactic simplification and text cohesion. Res. Lang. Comput. 4(1), 77–109 (2006)
3. Knight, K., Marcu, D.: Statistics-based summarization-step one: sentence compression. National Conference on Artificial Intelligence, 703–710 (2000)
4. Carroll, J., et al.: Simplifying text for language-impaired readers. Conference of the European Chapter of the Association for Computational Linguistics, 269–270 (1999)
5. Imu, K., et al.: Text simplification for reading assistance: a project note. Meeting of the Association for Computational Linguistics, 9–16 (2003)
6. Kaji, N., et al.: Verb paraphrase based on case frame alignment. Meeting of the Association for Computational Linguistics, 215–222 (2002)
7. Knight, K., Marcu, D.: Summarization beyond sentence extraction: a probabilistic approach to sentence compression. Am. J. Comput. Linguist. 19(1), 91–107 (2002)
8. Clarke, J., Lapata, M.: Models for sentence compression: a comparison across domains, training requirements and evaluation measures. Meeting of the Association for Computational Linguistics, pp. 377–384 (2006)
9. Filippova, K., Strube, M.: Dependency tree based sentence compression. International Conference on Natural Language Generation, 25–32 (2008)
10. Petersen, S.E., Ostendorf, M.: Text simplification for language learners: a corpus analysis. Symp. Lang. Appl. Technol. 69–72 (2007)
11. Gehring, J., et al.: A convolutional encoder model for neural machine translation. Meeting of the Association for Computational Linguistics, pp. 123–135 (2007)
12. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. Empirical Methods in Natural Language Processing, pp. 1724–1734 (2014)
13. Zhang, X., Lapata, M.: Sentence simplification with deep reinforcement learning. Empirical Methods in Natural Language Processing, pp. 584–594 (2017)
14. Vu, T., et al.: Sentence simplification with memory-augmented neural networks. North American Chapter of the Association for Computational Linguistics, pp. 79–85 (2018)
15. Zhang, L., Deng, H.: Sentence simplification based on multi-stage encoder model. IEEE Access. 7, 174246–174256 (2019)
16. Shen, Y., et al.: Ordered neurons: integrating tree structures into recurrent neural networks. pp. International Conference on Learning Representations (2019)
17. Xu, W., et al.: Problems in current text simplification research: new data can help. TACL 3, 283–297 (2015)
18. Vickrey, D., Koller, D.: Sentence simplification for semantic role labeling. Meeting of the Association for Computational Linguistics, pp. 344–352 (2008)
19. Woodsend, K., Lapata, M.: Learning to simplify sentences with quasysynchronous grammar and integer programming. Empirical Methods in Natural Language Processing, pp. 409–420 (2011)
20. Wubben, S., et al.: Sentence simplification by monolingual machine translation. Meeting of the Association for Computational Linguistics, pp. 1015–1024 (2012)
21. Narayan, S., Gardent, C.: Hybrid simplification using deep semantics and machine translation. Meeting of the Association for Computational Linguistics, pp. 435–445 (2014)
22. Curran, J.R., et al.: Linguistically motivated large-scale NLP with C&C and boxer. Meeting of the Association for Computational Linguistics, pp. 33–36 (2007)
23. Xu, W., et al.: Optimizing Statistical Machine Translation for Text Simplification. Transactions of the Association for Computational Linguistics 4, 401–415 (2016)
24. Sutskever, L., et al.: “Sequence to Sequence Learning with Neural Networks”, Neural Information Processing Systems, pp. 3104–3112 (2014)
25. Rush, A.M., et al.: A neural attention model for abstractive sentence summarization. Empirical Methods in Natural Language Processing, pp. 379–389 (2015)
26. Munkhdalai, T., Yu, H.: Neural Semantic encoders. Conference of the European Chapter of the Association for Computational Linguistics, pp. 397–407 (2017)
27. See, A., et al.: Get to the point: summarization with pointer-generator networks. Meeting of the Association for Computational Linguistics, pp. 1073–1083 (2017)
28. Guo, H., et al.: Dynamic multi-level multi-task learning for sentence simplification. International Conference on Computational Linguistics, pp. 462–476 (2018)

29. Nishihara, D., et al.: Controllable text simplification with lexical constraint loss. Meeting of the Association for Computational Linguistics, pp. 260–266 (2019)

30. Schulman, J., et al.: Gradient estimation using stochastic computation graphs. Advances in Neural Information Processing Systems, pp. 3528–3536 (2015)

31. Klein, G., et al.: Opennmt: open-source toolkit for neural machine translation. Meeting of the Association for Computational Linguistics, pp. 67–72 (2017)

32. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res. 12(July), 2121–2159 (2011)

33. Jean, S., et al.: Montreal neural machine translation systems for WMT’15. In: Proceedings of the Tenth Workshop on Statistical Machine Translation, pp. 134–140 (2015)

How to cite this article: Deng, C., Zhang, L., Deng, H.: Improving sentence simplification model with ordered neurons network. CAAI Trans. Intell. Technol. 1–10 (2021). https://doi.org/10.1049/cit2.12047