Character-Aware Low-Resource Neural Machine Translation with Weight Sharing and Pre-Training

Yichao Cao[1,2], Miao Li[1], Tao Feng[1,2], and Rujing Wang[1,2]

[1] Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China/P. R.China
{mli,rjwang}@iim.ac.cn
[2] University of Science and Technology of China, Hefei 230026, China/P. R. China
{cycao,ft2016}@mail.ustc.edu.cn

Abstract. Neural Machine Translation (NMT) has recently achieved the state-of-the-art in many machine translation tasks, but one of the challenges that NMT faces is the lack of parallel corpora, especially for low-resource language pairs. And the result is that the performance of NMT is much less effective for low-resource languages. To address this specific problem, in this paper, we describe a novel NMT model that is based on encoder-decoder architecture and relies on character-level inputs. Our proposed model employs Convolutional Neural Networks (CNN) and highway networks over character inputs, whose outputs are given to an encoder-decoder neural machine translation network. Besides, we also present two other approaches to improve the performance of the low-resource NMT system much further. First, we use language modeling implemented by denoising autoencoding to pre-train and initialize the full model. Second, we share the weights of the front few layers of two encoders between two languages to strengthen the encoding ability of the model. We demonstrate our model on two low-resource language pairs. On the IWSLT2015 English-Vietnamese translation task, our proposed model obtains improvements up to 2.5 BLEU points compared to the baseline. We also outperform the baseline approach more than 3 BLEU points on the CWMT2018 Chinese-Mongolian translation task.

Keywords: Low-resource Neural Machine Translation · Character-level · Weight sharing · Pre-training.

1 Introduction

Machine Translation (MT) is a challenging task in the field of natural language processing, which is aimed at transforming a source language into a target language automatically. Thanks to recent advances in deep learning, NMT has reached large improvements in some standard benchmarks and even has achieved near human-level performance on several language pairs [1,2]. Unfortunately, the performance of NMT depends heavily on numerous high-quality parallel corpora, which is only available for a few language pairs. In most cases, the NMT model has better performance in high-resource settings because of the help of significant amounts of training data. Some recent advances have reported poor performance of NMT systems in low-resource settings [3,4]. Therefore, improving the performance of low-resource NMT is a valuable study.

In this work, we propose a novel NMT model that exploits character information by a character-level CNN, whose output is used as an input to an encoder-decoder neural machine translation network module. Unlike previous works [5] that combine input word embeddings with map features from a character-level CNN module, our model only utilizes feature representations from CNN. Hence, we can take advantages of intra-word information and subword information to handle rare or out-of-vocabulary words of low-resource languages, especially when processing morphologically rich languages. Besides, our model no longer requires word vocabulary in source side by replacing word embedding layer with character-level CNN. The translation system of our model is based on the Transformer model [1], which follows an encoder-decoder architecture. It is based on attention mechanisms solely and dispenses with recurrence and convolutions entirely.

In order to improve the performance of our model even further, we present two approaches for low-resource neural machine translation. The first one is sharing weights between languages. More concretely, we employ the multi-task learning framework to build our NMT model, which is based on encoder-decoder architecture. We first build two NMT models in two opposite directions for a given low-resource language pair such as English-Vietnamese (e.g. English-to-Vietnamese and Vietnamese-to-English). Inspired by [6], we then share the weights of the front few layers of two encoders that are responsible for mapping input sentences into high-level representation space. Note that, two decoders are not shared. Intuitively, the shared encoder can make better use of the similarity and complementarity between different languages, especially for low-resource language pairs. The second approach utilizes strong language modeling to pre-train and initialize our full model, which can enhance the quality of translation results for low-resource NMT. Similar to [7], we apply language modeling pre-training via training the encoder-decoder system as a denoising autoencoder [8]. Furthermore, denoising autoencoding is still in progress during translation training. To summarize, our contributions are as follows:

- We propose a character-aware and weight-sharing constraint for low-resource NMT, and at the same time apply language modeling which is implemented by denoising autoencoding to pre-train and initialize the entire model. To maintain the internal characteristics of each language for the model, we also keep denoising autoencoding throughout the training.
- We conduct extensive experiments on English-Vietnamese and Chinese-Mongolian low-resource translation tasks. The experiment results demonstrate that the proposed approach is effective for low-resource NMT and significantly outperforms the baseline model.

2 Background and Related Work

In recent years, a neural machine translation model is usually implemented by an encoder-decoder architecture [9], and the encoder and decoder of the NMT model are often based on recurrent neural networks (RNN) or Transformer. The encoder reads a source sentence $X = (x_1, \ldots, x_{T_x})$ as an input and encodes it into a high-level representation $H = (h_1, \ldots, h_{T_x})$. Then the decoder generates corresponding translation $Y = \left(y_1, \ldots, y_{T_y}\right)$ based on the encoded sequence of hidden states H, where x_t, h_t and y_t are the symbols of source language, hidden states and target language, respectively. Generally, the NMT model is trained to maximize the conditional log-probability of a target sentence Y given a source sentence X as:

$$p\left(Y \mid X\right) = \prod_{t=1}^{T} \left(Y_t \mid Y_{0:t-1}, X; \theta\right) \tag{1}$$

where θ is the parameters of the model, and T represents the length of Y. In this work, our model is based on the Transformer model, which uses only attention mechanisms. Attention mechanism can be described as mapping queries (Q) and a set of key-value (K-V) pairs to outputs, where Q, K and V are all vectors. In practice, we compute the attention function as follows:

$$R_a = softmax\left(QK^T\right)V \tag{2}$$

where R_a denotes the attention vector.

In the past, many studies have produced competitive results on low-resource NMT. [3] introduced a transfer learning approach to improve the performance of low-resource NMT, and [10] took advantage of a model-agnostic meta-learning algorithm to train the model. Other attempts exploited the availability of monolingual corpora to obtain better results on low-resource NMT [11,12]. However, most of the NMT models belong to a family of word-level systems, whose large vocabulary is often filled with many similar words. Besides, many words are out-of-vocabulary (OOV) because of the paucity of data for low-resource language pairs. In our approach, character-level NMT and language modeling pre-training are employed to address the above problems. There have been numerous great efforts at character-level neural machine translation. [13] made attempt to compose words from individual characters with the help of Long Short Term Memory (LSTM). [14] exploited character-aware word vectors to replace the standard word representations on the source side and demonstrated it is much effective. There is also a line of work on character-level NMT that processes words at the character level [15,16]. For language modeling pre-training, the Generative Pre-trained Transformer (OpenAI GPT) [17] and the Bidirectional Encoder Representations from Transformers (BERT) [7] are trained on the downstream tasks by fine-tuning the pre-trained language modeling parameters. Our work is most aligned with the language modeling proposed by [18], which made use of denoising autoencoding to train models on both source and target languages.

A recent trend in multi-task neural machine translation is to share the weights of the model between different languages. [6,19] investigated how to leverage

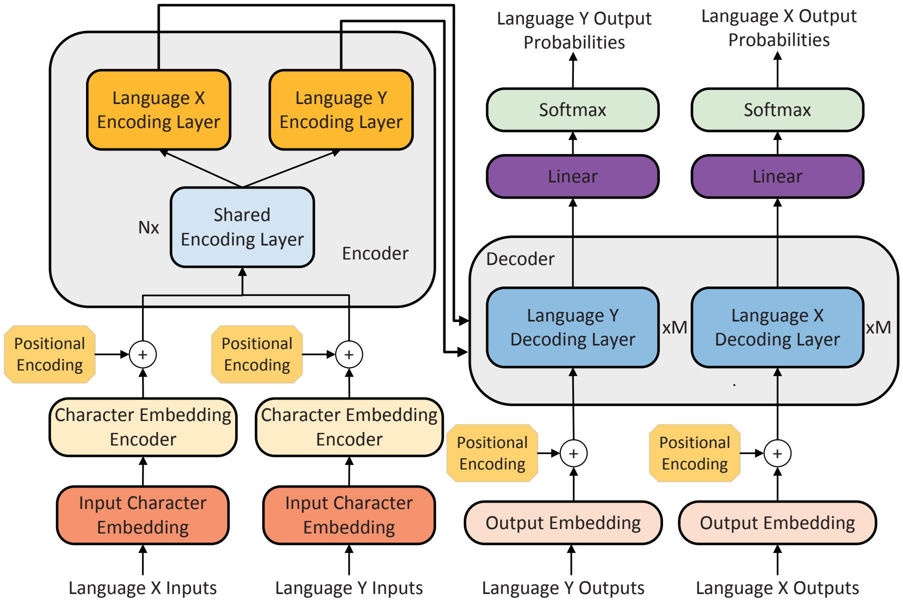
Language X Encoding Layer
Language Y Encoding Layer
Shared Encoding Layer
Nx
Encoder
Positional Encoding
Positional Encoding
Character Embedding Encoder
Character Embedding Encoder
Input Character Embedding
Input Character Embedding
Language X Inputs
Language Y Inputs
Language Y Output Probabilities
Language X Output Probabilities
Softmax
Softmax
Linear
Linear
Decoder
Language Y Decoding Layer
xM
Language X Decoding Layer
xM
Positional Encoding
Positional Encoding
Output Embedding
Output Embedding
Language Y Outputs
Language X Outputs

Fig. 1. The architecture of the proposed model. The model uses the multi-task learning framework for two languages X, Y of a given low-resource language pair. In source side, the standard word input embedding is replaced by a combination of character-level CNN and highway network which is called Character Embedding Encoder. In target side, we still use the standard word embedding lookup table. We share N layers of the encoder but do not share any layers of the decoder. The feed forward layers and the softmax layers are employed to get output probabilities of two languages.

weight sharing in unsupervised and supervised NMT to improve the quality of the translation results. Our proposed model also belongs to this scenario, which can exploit the similarity and complementarity between different languages and mitigate the overfitting issues for low-resource NMT.

3 Model Architecture

It is well known that low-resource NMT systems usually suffer from the limited amount of parallel data, OOV words and model overfitting. To address these specific problems, we propose a character-aware low-resource NMT model that follows the encoder-decoder architecture. As shown in Fig. 1, we introduce the multi-task learning framework to build the whole model and use the data of one low-resource language pair as the model inputs. The character-level inputs of different languages are fed into the combination of CNN and highway network that is called Character Embedding Encoder (CEE). The machine translation encoder-decoder module of our model is based on the Transformer model, and the input of encoder is obtained from CEE. Note that we provide the standard word-level embeddings as the decoder inputs rather than character-level, and predictions are also still at the word-level.

3.1 Character Embedding Encoder

In recent years, NMT has benefited from character-aware representations on the source side or target side [14,20]. Inspired by [21], we replace the word embedding layer with a combination of CNN and highway network [22] which called CEE to acquire the embedding of a word. The structure of CEE is illustrated in Fig.2, and the input of the model encoder for each word is the output from CEE. Let $X = [x_1, \ldots, x_n]$ be the input sentence of our model, where x_i is i-th word of sentence. And assume that word x_i consists of a sequence of characters $\left[x_c^1, \ldots, x_c^m\right]$, where m is the length of word x_i. Let $\mathbf{c_i}$ be the embedding of a character, then character embeddings are concatenated together as input $\mathbf{C_{x_i}}$ of CNN for each word x_i. A narrow 1D convolution is applied between concatenation of character embeddings $\mathbf{C_{x_i}}$ and a kernel $K \in \mathbb{R}^{d_c \times w}$ with width w, where d_c denotes the dimensionality of character embeddings. Then we utilize max-over-time pooling to keep only the maximum value of the output for each convolutional filter, and concatenate these feature maps as the input Z_f of the highway network.

To acquire a better embedding for word x_i, we employ the highway networks and a fully connected layer on Z_f. A highway network allows some character n-grams to be combined to build new features, and other character n-grams to remain primitive features. We can compute a new set of features Z with Z_f by the following formula,

$$Z = u \odot g\left(W_H Z_f + b_H\right) + u' \odot Z_f \tag{3}$$

where $u = \sigma\left(W_u Z_f + b_u\right)$ is called the $transform$ gate, $u' = 1 - u$ is called the *carry* gate, g denotes a nonlinearity, and σ, W_H, W_u, b_H, b_u are all trainable parameters. Besides, a fully connected feed-forward layer with a *ReLU* activation is applied to seek a more satisfying word representation for each word, and calculated as follows:

$$E_w = ReLU\left(Z W_{ff}\right) + b_{ff} \tag{4}$$

where E_w is the final word embedding of the word x_i, W_{ff} and b_{ff} denote the transformation matrix and bias respectively. The dimensionality of E_w is d_m, which matches with the dimension of model encoder and decoder hidden layers.

3.2 Weight Sharing Encoder

It is well known that there are the similarity and complementarity between different languages, [23] has demonstrated that multilingual translation is indeed strongly beneficial for resource-scarce language pairs. Our proposed model takes advantage of the multi-task learning framework, and employs jointly training for two languages in a given low-resource language pair.

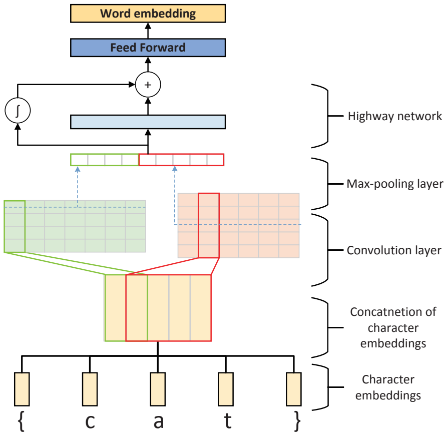

Fig. 2. The architecture of Character Embedding Encoder (CEE). Firstly, combining character embeddings of a word into a representation **C**. Then convolution operations are applied between **C** and multiple filter matrices. We only show 2 convolution filters in the above illustration. A max-over-time pooling operation is employed to get a high-level representation **Z** of the word, and **Z** will be used as an input to the highway network. Finally, a fully connected feed-forward layer is applied over the output of the highway network to achieve the embedding of this word.

Specifically, for example, we build two encoder-decoder translation systems for English-Vietnamese whose parallel corpus is not sufficient. One of them is trained to translate from English to Vietnamese, and the other is trained in the opposite direction (Vietnamese to English). And then we share the weights of the front few layers of two encoders, but not share the weights of two decoders. Intuitively, encoder weight sharing can make the best of the similarity and complementarity between two languages to extracting the high-level representation of each input. Furthermore, the shared encoder may keep the balance between two languages of a given low-resource language pair, and do not cause overfitting caused by the data scarcity problem. Note that we do not share the last layer of two encoders and all layers of two decoders in our approach, because the separate encoder or decoder preserves the uniqueness and internal characteristics of each language. Another reason we do not share decoders of two translation systems is that the decoder utilizes not only self-attention mechanism but also dot-product attention[1]. The alignment matrix calculated from the dot-product attention mechanism is not same for different source and target languages, therefore we utilize two independent decoders to building the NMT model.

3.3 Denoising Autoencoding and Training

Language modeling pre-training has achieved great success for most natural language processing tasks, and [24] also showed the effectiveness of cross-lingual

Algorithm 1 Training for our proposed model

```
Input: Monolingual data $D'_x$, $D'_y$ and parallel data $D_x$, $D_y$
Output: An NMT model: $M_{x\to y, y\to x}$
for i=1 to N do
    Sample data $X'\,(i) \in D'_x$ and $Y'\,(i) \in D'_y$
    Add noise to $X'\,(i)$, $Y'\,(i)$ and obtain noised inputs $X_c{}'\,(i)$, $Y_c{}'\,(i)$
    Train character-aware language modeling $L_x$ and $L_y$ via denoising autoencoding using $X_c{}'\,(i)$ and $Y_c{}'\,(i)$
Initialize the translation model $M^{(0)}_{x\to y, y\to x}$ using $L_x$ and $L_y$
for t=1 to T do
    Train $M^{(t)}_{x\to y, y\to x}$ using $D_x$ and $D_y$
    Fine-tune $M^{(t)}_{x\to y, y\to x}$ via denoising autoencoding using $D'_x$ and $D'_y$
return $M^{(T)}_{x\to y, y\to x}$
```

pre-training on unsupervised machine translation. In our method, language modeling is accomplished via denoising autoencoding, that can reconstruct the original sentences from the noised inputs and learn some useful structures in the data. To be specific, let $X' = [x'_1, \ldots, x'_n]$ be the original sentence sampled from monolingual data D'_x, we add three different types of noise to X' and get the noised input X'_c. Firstly, we shuffle the input sentence with a random permutation γ and verify the condition:

$$|\gamma\,(i) - i| \leq W', \forall i \in \{1, n\} \tag{5}$$

where W' is a tunable parameter. Secondly, we delete some word of X' with a probability P_d. In practice, we consider $P_d = 0.1$. Last but not least, we sample randomly 10% of the word from X', and replace them by a special token '<*mask*>' 80% of the time and keep them unchanged 20% of the time similar to [7]. This way, the encoder is forced to keep a distributional contextual representation of X', and the decoder is trained to learn the internal structure of the sentence to generate the original inputs.

In our work, we first pre-train the language modeling for N step by minimizing:

$$\mathcal{L}_{ae} = \mathbb{E}_{X' \sim D'_x}\left[-logP_{x\to x}\left(X' \mid X'_c\right)\right] + \mathbb{E}_{Y' \sim D'_y}\left[-logP_{y\to y}\left(Y' \mid Y'_c\right)\right] \tag{6}$$

where X'_c and Y'_c denote noised inputs, D'_x and D'_y are monolingual data, $P_{x\to x}$ and $P_{y\to y}$ are the denoising autoencoder both operating on two languages, respectively.

As we described, the proposed model is based on the multi-task learning framework. Hence, joint training is applied to a given low-resource language pair translation, and the NMT model is initialized using the pre-trained model after finishing N step pre-training. Then we can train our NMT model in two opposite directions for one language pair, and fine-tune it through denoising autoencoding at every iteration. In practice, it is worth noting that the proportion of denoising autoencoding loss in the total loss is decreased along with the training. The

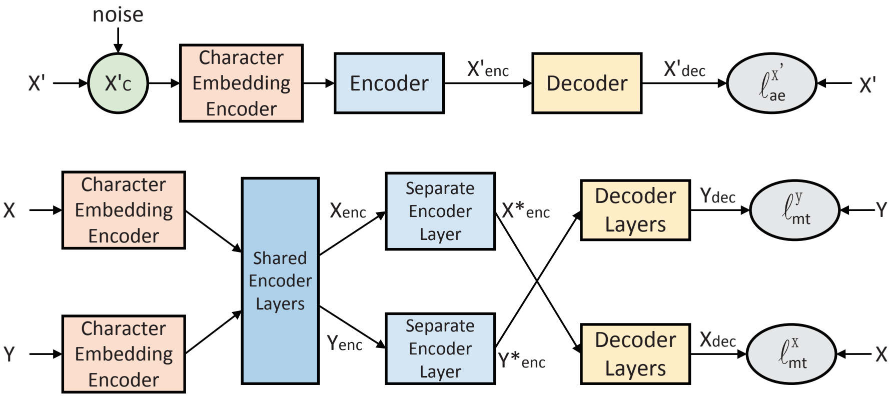

Fig. 3. Illustration of denoising autoencoding and translation procedure. Top (denoising autoencoding): the model learns to reconstruct the original sentence X' from the noised sentence X'_c. X'_{enc} and X'_{dec} are the output of the encoder and decoder, respectively. Bottom (translation): as described before, we jointly train the translation model for a given low-resource language pair. X and Y are source inputs, X_{enc}, Y_{enc} and X^*_{enc}, Y^*_{enc} are outputs of shared encoder layers and separate encoder layer respectively. X_{dec} and Y_{dec} are decoder layers outputs. The gray ellipses indicate terms in the loss function.

overview of our algorithm is given in Algorithm 1. Let X and Y be sentences sampled from parallel data D_x and D_y, the translation loss can be written as:

$$\mathcal{L}_{mt} = \mathbb{E}_{X\sim D_x, Y\sim D_y}\left[\left(-logP_{x\to y}\left(Y \mid X\right)\right) + \left(-logP_{y\to x}\left(X \mid Y\right)\right)\right] \tag{7}$$

where $P_{x\to y}$ or $P_{y\to x}$ denotes the translation model.

In summary, as shown in Fig 3, the final objective function at one iteration of our method is thus:

$$\mathcal{L}_{model} = \lambda_{ae}\mathcal{L}_{ae} + \lambda_{mt}\mathcal{L}_{mt} \tag{8}$$

where λ_{ae} and λ_{mt} are hyper-parameters weighting the importance of the denoising autoencoding and translation loss. λ_{ae} is decreased along with the training.

4 Experiments

In this section, we first describe the datasets and experimental protocol that we used. Then, we compare the proposed approach with the baseline method and other methods. The results and related discussions will be reported in the end.

4.1 Datasets

In our experiments, we consider two low-resource language pairs: English-Vietnamese and Chinese-Mongolian. For English-Vietnamese, we utilize available parallel corpus from IWSLT 2015 which is composed of 131K sentence pairs. For

Chinese-Mongolian, we use CWMT 2018 Chinese-Mongolian dataset consisting of about 260K sentence pairs. We separate all parallel sentences as the monolingual corpora. We measure the performance of all methods by BLEU score, which is often used in translation tasks. We report BLEU scores on tst2012 and tst2013 for English-Vietnamese, and report results on test2017 for Chinese-Mongolian.

Moreover, we apply word segmentation to the Chinese sentences using THU Lexical Analyzer for Chinese (THULAC) [25]. Because our model is a character-to-word (*char2word*) translation model, we only need the character vocabulary for the *source-side* inputs. In practice, we give a source character vocabulary of size 200-300 for each language except Chinese. For Chinese character vocabulary, we take the most frequent five thousand Chinese characters and replace the rest with '<*unk*>' token. Like most word-level NMT systems, the standard word vocabulary is employed on the target side.

4.2 Training Details

In this study, we built the proposed model upon the character-level CNN, highway networks and Transformer cells. Following [21], the CNN has filters of width [1, 2, 3, 4, 5, 6, 7] of size [50, 100, 150, 200, 200, 200, 200] for a total of 1100 filters, and the highway network has 2 layers with a *ReLU* activation. Besides, the dimensionality of character embeddings is set to 50. Our encoder-decoder translation architecture is based on Transformer cells with 2048 hidden units and 8 heads. We use 6 layers both in the encoder and decoder, and share the weights of the front 5 layers of the encoder between two source languages. The dimensionality of the word embeddings and the hidden layers is set to 512. The model is trained using the Adam optimizer [26] with $\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-9}$, and the initial learning rate is set to 2. The learning rate varies according to the formula:

$$learning\_rate = d_m^{-0.5} \cdot min\left(gs \cdot ws^{-1.5}, gs^{-0.5}\right) \tag{9}$$

where d_m is the dimensionality of the hidden layers, gs and ws denote the global step and warmup step, respectively. We also apply dropout [27] with a rate of 0.1 to the output of each sub-layer, the embeddings and the positional encodings.

To evaluate the effectiveness of the proposed method and the importance of different components, we compare our model with the baseline method Transformer and other models. We utilize the symbol *CWSP* to denote our proposed model, and the following methods will be compared:

- Transformer (word2word): This approach is our baseline method similar to [1].
- Transformer (char2word): This model is the same as the baseline method except for the source inputs, which is character-level.
- CWSP (without LM): This is our model without denoising autoencoding.
- CWSP (4 shared layers): This is our approach which only shares four layers of the encoder.
- CWSP (6 shared layers): This is our model which shares all six layers of the encoder.

Table 1. The translation performance on IWSLT2015 English-Vietnamese tst2012 set, tst2013 set and CWMT2018 Chinese-Mongolian test2017 set.

Models	en-vi (tst2012)	vi-en (tst2012)	en-vi (tst2013)	vi-en (tst2013)	zh-mn (test2017)	mn-zh (test2017)
Transformer (word2word)	24.69	22.36	26.94	24.42	29.14	15.87
Transformer (char2word)	24.77	22.75	27.71	24.65	30.66	16.33
CWSP (without LM)	24.95	23.60	27.84	25.58	31.20	16.55
CWSP (4 shared layers)	25.29	24.23	27.78	26.46	30.94	16.37
CWSP (6 shared layers)	25.08	24.12	27.79	26.59	31.25	18.69
CWSP (our full model)	**25.41**	**24.40**	**27.93**	**26.93**	**31.92**	**18.86**

- CWSP (our full model): This is our full model which shares 5 layers of the encoder and does not share any layers of the decoder.

4.3 Results and Analysis

Table 1 reports the translation BLEU scores of different systems on English-Vietnamese and Chinese-Mongolian tasks. The results show that our proposed approach obtains significant improvements compared with the baseline method. Our model can achieve at least +0.7 BLEU points improvement in English-to-Vietnamese translation, and up to +2.5 BLEU points in Vietnamese-to-English translation. We also can reach 31.92 and 18.86 BLEU scores in Chinese-Mongolian test2017, outperforming the baseline method by +2.78 and +2.99 points respectively. As can be seen, the proposed approach is effective for low-resource machine translation.

Besides, compared to the baseline model, the Transformer (char2word) model leads to improvements of up to +1.52 BLEU on two translation tasks. This confirms the CEE module of our model is able to encode semantically meaningful features, and then generates better word embeddings. It has a beneficial effect on low-resource NMT. The approach named *CWSP*(without LM) gets improvements compared with the Transformer (char2word) model, with up to +0.93 BLEU points on English-Vietnamese translation task. It reveals that the model can encode the inputs better via sharing the weights of layers of encoders between two languages. And the encoder of our model can utilize the similarity and complementarity of two different languages to enhance the low-resource neural machine translation performance. We also find that the shared encoder is less prone to overfitting in practice.

Our full model obtains improvements up to +1.35 BLEU points compared to the proposed model without denoising autoencoding (*CWSP*(without LM)) on English-Vietnamese test sets, and at least +0.72 BLEU points on Chinese-Mongolian test set. It shows that pre-training allows the NMT model to learn the semantic features of the language, and guides the decoder to generate more satisfactory sentences. We also investigate how the number of weight sharing layers of encoders affects translation performance. From Table 1, we find that better performance is achieved when 5 layers are shared in our model. Therefore, we set the number of the encoder layers as 5 in our full model. Our model with 4 shared layers obtains up to –0.47 points decline than our full model in English-Vietnamese translation, and at least -0.98 points in Chinese-Mongolian translation. It verifies less shared layers of two encoders do not take full advantage of the relationship between two languages of a given resource-scarce language pair. The BLEU score is worse than the result of our full model when we share all of the six layers of two encoders. And we also share some layers of two decoders, the results are even worse than the baseline model. We explain these as that the shared encoder or the shared decoder are weak in keeping the unique characteristics of each language of given low-resource language pairs. The separate encoder and decoder can extract the distinctive characteristics for different languages, thus we share the front few layers of two encoders rather than all layers, and do not share any decoders.

5 Conclusion and Future Work

In this work, we introduce a character-aware encoder-decoder architecture for low-resource neural machine translation, and two strategies are employed to improve the translation performance for resource-scarce language pairs. On the one hand, we make use of weight sharing to enhance the encoder of the NMT model, which can also reduce overfitting. On the other hand, we initialize our model using the pre-trained model which is implemented by denoising autoencoding. Meanwhile, the model is fine-tuned via denoising autoencoding during training. The experiments show that the proposed model outperforms the baseline method and is effective for low-resource language pairs such as English-Vietnamese and Chinese-Mongolian.

In the next stage of the study, we may investigate how to apply character aware to the unsupervised NMT model. Furthermore, we will also research the performance of our model on some large-scale parallel corpora.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 61572462 and the 13th Five-year Informatization Plan of Chinese Academy of Science, Grant No. XXH13505-03-203.

References

1. Vaswani, A., Shazeer, N.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
2. Hassan, H., Aue, A., et al.: Achieving human parity on automatic chinese to english news translation. arXiv preprint arXiv:1803.05567 (2018)
3. Zoph, B., Yuret, D.: Transfer learning for low-resource neural machine translation. arXiv preprint arXiv:1604.02201 (2016)
4. Koehn, P., Knowles, R.: Six challenges for neural machine translation. arXiv preprint arXiv:1706.03872 (2017)

5. Santos, C.N.d., Guimaraes, V.: Boosting named entity recognition with neural character embeddings. arXiv preprint arXiv:1505.05008 (2015)
6. Yang, Z., Chen, W.: Unsupervised neural machine translation with weight sharing. arXiv preprint arXiv:1804.09057 (2018)
7. Devlin, J., Chang, M.W.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Vincent, P., Larochelle, H.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. pp. 1096–1103. ACM (2008)
9. Bahdanau, D., Cho, K.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
10. Gu, J., Wang, Y.: Meta-learning for low-resource neural machine translation. arXiv preprint arXiv:1808.08437 (2018)
11. Sennrich, R., Haddow, B.: Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709 (2015)
12. Currey, A., Barone, A.V.M.: Copied monolingual data improves low-resource neural machine translation. In: Proceedings of the Second Conference on Machine Translation. pp. 148–156 (2017)
13. Ling, W., Trancoso, I.: Character-based neural machine translation. arXiv preprint arXiv:1511.04586 (2015)
14. Ruiz Costa-Jussà M, Rodríguez Fonollosa J A. Character-based neural machine translation[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. pp. 357-361 (2016).
15. Luong, M.T., Manning, C.D.: Achieving open vocabulary neural machine translation with hybrid word-character models. arXiv preprint arXiv:1604.00788 (2016)
16. Passban, P., Liu, Q., Way, A.: Improving character-based decoding using targetside morphological information for neural machine translation. arXiv preprint arXiv:1804.06506 (2018)
17. Radford, A., Narasimhan, K.: Improving language understanding with unsupervised learning. Tech. rep., Technical report, OpenAI (2018)
18. Lample, G., Ott, M.: Phrase-based & neural unsupervised machine translation. arXiv preprint arXiv:1804.07755 (2018)
19. Firat, O., Cho, K.: Multi-way, multilingual neural machine translation with a shared attention mechanism. arXiv preprint arXiv:1601.01073 (2016)
20. Renduchintala, A., Shapiro, P.: Character-aware decoder for neural machine translation. arXiv preprint arXiv:1809.02223 (2018)
21. Kim, Y., Jernite, Y.: Character-aware neural language models. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
22. Srivastava, R.K., Greff, K.: Training very deep networks. In: Advances in neural information processing systems. pp. 2377–2385 (2015)
23. Lee, J., Cho, K.: Fully character-level neural machine translation without explicit segmentation. Transactions of the Association for Computational Linguistics 5, 365–378 (2017)
24. Lample, G., Conneau, A.: Cross-lingual language model pretraining (2019)
25. Sun, M., Chen, X.: Thulac: An efficient lexical analyzer for chinese. Tech. rep., Technical Report. Technical Report (2016)
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
27. Srivastava, N., Hinton, G.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)