A Character-Aware Encoder for Neural Machine Translation

Zhen Yang, Wei Chen*, Feng Wang, Bo Xu
Institute of Automation, Chinese Academy of Sciences
No.95 Zhongguancun East Road
{yangzhen2014, wei.chen.media, feng.wang, xubo}@ia.ac.cn

Abstract

This article proposes a novel character-aware neural machine translation (NMT) model that views the input sequences as sequences of characters rather than words. On the use of row convolution (Amodei et al., 2015), the encoder of the proposed model composes word-level information from the input sequences of characters automatically. Since our model doesn't rely on the boundaries between each word (as the whitespace boundaries in English), it is also applied to languages without explicit word segmentations (like Chinese). Experimental results on Chinese-English translation tasks show that the proposed character-aware NMT model can achieve comparable translation performance with the traditional word based NMT models. Despite the target side is still word based, the proposed model is able to generate much less unknown words.

1 Introduction

Neural machine translation conducts end-to-end translation with a source encoder and a target decoder, producing promising results (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014). With the emerging of the attention-based encoder-decoder model, NMT has achieved comparable even better translation performance with the traditional statistical machine translation (SMT) (Bahdanau et al., 2014; Ranzato et al., 2015; Shen et al., 2015; Tu et al., 2016). The success of NMT lies in its strong ability of composing the global context information. However, as a newly approach, the NMT model has some flaws and limitations that may jeopardize its translation performance (Luong et al., 2014; Sennrich et al., 2015; He et al., 2016). One of the most glaring limitations is that the NMT model is weak in handling the rare and out-of-vocabulary (OOV) words, since the NMT system usually uses the top-N (30000-50000) frequent words in the training corpus and regards other words as unseen words (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2014; Cohn et al., 2016). Two different kinds of approaches have been proposed to handle the OOV problem in NMT: ***vocabulary-specific*** approaches and ***unit-specific*** approaches.

The ***vocabulary-specific*** approaches seek to cover more words by using a larger vocabulary (Mnih and Kavukcuoglu, 2013; Cho et al., 2015) or using an identity translation dictionary in a post-processing step (Luong et al., 2014). Intuitively, these approaches can alleviate the OOV problems to a certain extent only if the vocabulary can be expanded large enough. However, these approaches are incapable of solving the OOV problems completely since the vocabulary is always limited.

As opposed to ***vocabulary-specific*** approaches, the ***unit-specific*** approaches try to use more fine-grained processing units than words, like sub-word units (Sennrich et al., 2015) or even character-level units (Ling et al., 2015b; Chung et al., 2016). Regarding the character as the basic processing unit is a new trend in the field of NLP and the character-level models have been widely used in NLP tasks (Ling et al., 2015a; Zhang et al., 2015; Golub and He, 2016). Developing character-level NMT models is attractive for multiple reasons. Firstly, it opens the possibility for models to generate unseen source words, since each word can be composed from different characters. Secondly, the vocabulary size of the

*Wei Chen is the corresponding author of this paper

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3063–3070, Osaka, Japan, December 11-17 2016.

model can be reduced dramatically as only the characters need to be modeled explicitly. This enables the character-level NMT model to solve many scalability issues, both in terms of computational speed and memory requirements. Finally, as each character occurs frequently in the training corpus, all of the character embeddings are able to get full trained. Hence they represent their corresponding characters very well. However, in the word-based NMT, the word embeddings for rare words, which hardly occur in the training corpus, are absent of enough training. Based on the state-of-the-art attention based encoder-decoder framework, some character-level NMT models have been proposed recently. (Chung et al., 2016) focus on representing the target side as a character sequence with a bi-scale recurrent neural network. In (Ling et al., 2015b), a character-based word representation model is proposed in the source side. (Luong and Manning, 2016) proposes a hybrid architecture for NMT that translates mostly at the word level and consults the character components for rare words when necessary. Most of the works mentioned above apply the bidirectional RNN to compose the word representation from its characters. Hence, its necessary to know the boundary between two words beforehand. These models are applicable for languages in which the words are segmented with explicit boundaries, such as English, French and etc.

In this work, we propose a novel character-aware NMT model that learns to encode at the character level. On the use of row convolution, the proposed model can be applied to the language which has no explicit word segmentations, like Chinese. We still represent the target side as a sequence of words. This paper has two main contributions:

- We propose a simple and novel NMT model which views the input as sequences of characters. We firstly rule out the word segmentation processing step for languages without explicit word segmentations.
- We introduce several different row convolution methods and investigate their effectiveness in NMT. Row convolution is a newly-emerged technique and has shown its great effectiveness in speech recognition (Amodei et al., 2015).

Experimental results show that contrarily to previous belief, the proposed character-aware NMT model can generate results on par with the word-based NMT models. The rest of this paper is organized as follows. Section 2 describes related works. In Section 3, we propose our character-aware NMT model. Experiments and results are described in Section 4. We conclude in Section 5.

2 Related works

In this section, we describe the basis of this work: the attention-based encoder-decoder NMT model and the row convolution method.

2.1 Attention-based encoder-decoder

This subsection briefly describes the attention-based NMT (RNNsearch) (Bahdanau et al., 2014), on which the character-aware NMT model is built. The RNN search model simultaneously conducts dynamic alignment and generation of the target sentence and it produces the translation sentence by generating one target word at every time step. Given an input sequence $\boldsymbol{x} = (x_1, \ldots, x_{T_x})$ and previous translated words $(y_1, \ldots, y_{i-1})$, the probability of next word y_i is:

$$p(y_i|y_1, \ldots, y_{i-1}, \boldsymbol{x}) = g(y_{i-1}, s_i, c_i) \tag{1}$$

where s_i is an decoder hidden state for time step i, which is computed as:

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{2}$$

Here f and g are nonlinear transform functions, which can be implemented as long short term memory network(LSTM) or gated recurrent unit (GRU), and c_i is a distinct context vector at time step i, which is

calculated as a weighted sum of the input annotations h_j:

$$c_i = \sum_{j=1}^{T_x} a_{i,j} h_j \tag{3}$$

where h_j is the annotation of x_j from a bidirectional RNN. The weight a_{ij} for h_j is calculated as:

$$a_{i,j} = \frac{exp(e_{ij})}{\sum_{t=1}^{T_x} exp(e_{i,t})} \tag{4}$$

where

$$e_{i,j} = v_a tanh(W s_{i-1} + U h_j) \tag{5}$$

The mechanism of attention in RNN search makes the decoder focus on relevant words in the source sentence when generating the target word. The graphical illustration of the RNN search model is depicted in Fig.1(a).

2.2 Row convolution

(Amodei et al., 2015) firstly proposes the row convolution, which is used to look forward for a small portion of future information at the current time-step. Suppose at time-step t, the input h_t is a d-dimensional continuous vector and a future context of τ steps is considered. The model gets a feature matrix $h_{t:t+\tau}$ of size $d \times (\tau + 1)$. A parameter matrix of the same size as $h_{t:t+\tau}$ is defined as W. The activation r_t at the time-step t is computed as:

$$r_{t,i} = \sum_{j=1}^{\tau+1} W_{i,j} h_{t+j-1,i}, for 1 \leq i \leq d \tag{6}$$

Since the convolution-like operation in Eq.6 is row oriented for both W and $h_{t:t+\tau}$, it is called row convolution.

3 The character-aware NMT model

In this section, we describe the proposed character-aware NMT model in detail. Fig.1(b) is the graphical illustration of the proposed model. The basic architecture is a character-level encoder, which composes the input character embedding and its context embedding into the corresponding word-level representation.

3.1 Model overview

In a slightly generalized sense, the proposed character-aware NMT model is still an encoder-decoder. The encoder transformed the source sequence into the vector representation, which is then read by decoder to generate the output sequence.

Encoder The encoder in the proposed model is utilized in the character level. Specifically, the model is designed to compose the word-level information from the input sequence of characters. Compared to the RNN-encoder in (Bahdanau et al., 2014), there are two important differences

- **Context Computing** The proposed model builds a row convolution layer to compute the context vector for the current input character. The context vector preserves the context information which guides how to compose the word-level information.
- **Character composing** A character composing layer is used to compose the word-level information from the current input character and its corresponding context vector. The word-level information is then fed as input to the bidirectional RNN.

Decoder An RNN that reads the hidden variables of the encoder and predicts the target sequence. It is almost the same with the canonical RNN-decoder in (Bahdanau et al., 2014).

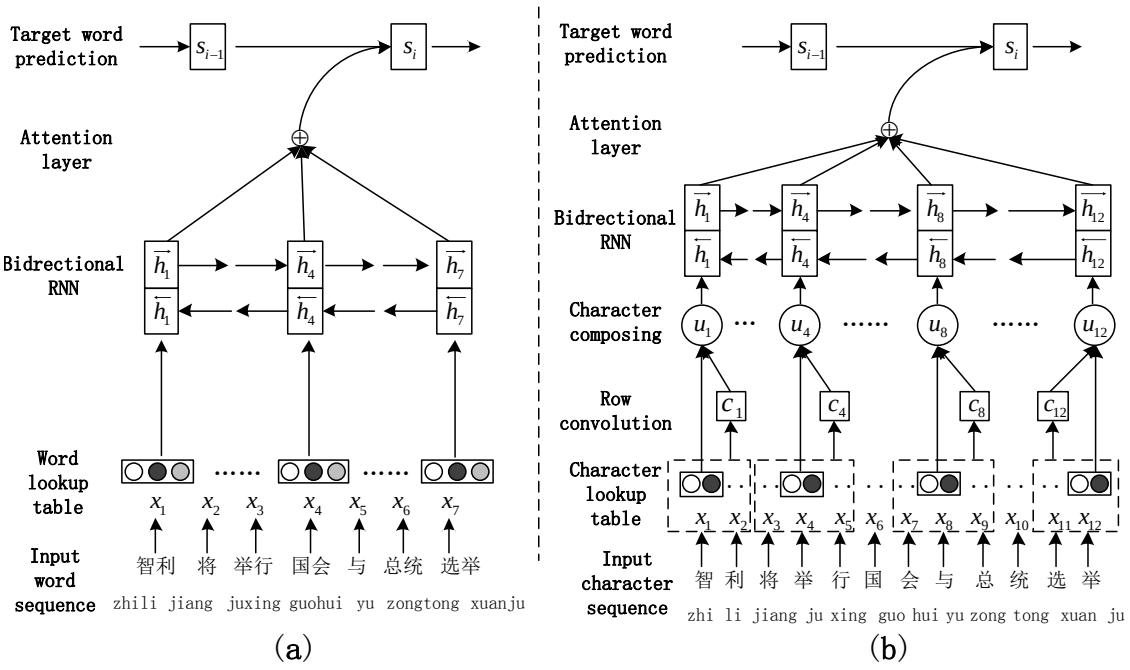

Figure 1: The graphical illustration of the proposed model. (a) is the traditional attention-based encoder-decoder model proposed by (Bahdanau et al., 2014). (b) is the proposed character-aware NMT model in this paper. Both of the two models try to translate the Chinese sentence "zhi li jiang ju xing guo hui yu zong tong xuan ju". The traditional attention-based model needs to segment the input sentence into Chinese words first. However, our character-aware model encodes the characters of the input sentence directly.

3.2 Bidirectional and concatenated row convolution for context

Given an input sequence of characters, the model projects each character into a continuous d-dimensional character vectors x_i using a character lookup table, which is similar to the word lookup table in the word-based NMT. Then, it builds a bidirectional and concatenated row convolution layer to compute the context vector c_i for x_i. Different from the traditional row convolution proposed in (Amodei et al., 2015) which only looks forward for the future context, the proposed bidirectional row convolution also looks behind to the history context. The intuition behind this layer is that, in additional to the future context, a small portion of history context is also needed to compose a full word-level representation for the character in current context. Suppose the input character x_i at time-step i, and the window size of the bidirectional row convolution is set as τ, we get a context matrix $x_{i-\tau,i+\tau}$ of size $d \times (2\tau + 1)$. We define a convolution matrix W of the same size as $x_{i-\tau,i+\tau}$. The activation c_i for the layer at time-step i is computed as:

$$c_i = [v_{i-\tau}; v_{i-\tau+1}; \ldots; v_{i+\tau}] \tag{7}$$

where $v_{i-\tau+t}$ is computed as:

$$v_{i-\tau+t} = w_t \times x_{i-\tau+t} (1 \leq t \leq (2\tau + 1)) \tag{8}$$

Since c_i is concatenated from $v^{i-\tau:i+\tau}$, the row convolution proposed in this paper is referred to as concatenated row convolution. For comparison and clarity, we call the row convolution in (Amodei et al., 2015) as summed row convolution.

3.3 Character composing

To fully utilize the character embedding x_i and its context vector c_i, we propose two different structures to compose the word-level representation u_i.

Forward character composing The forward character composing simply uses a linear transformation to combine the character embedding and its context embedding. For each character x_i, the corresponding word-level representation u_i is computed as:

$$u_i = M_1 x_i + M_2 c_i + b \tag{9}$$

where the weight matrix $M_1 \in R^{d \times d}$, the transformation matrix $M_2 \in R^{2\tau+1}$ convert the context c_i into the same dimension with x_i, the bias vector $b \in R^d$. Hence, the word-level representation u_i is kept the same dimension. In this model, the forward character composing layer is expected to learn the word-formation from the character and its neighboring characters automatically.

Recurrent character composing The recurrent character composing considers the interference from the former word-level representation u_{i-1} when computes u_i. Since the character x_{i-1} shares part of the neighboring characters with x_i, the context c_{i-1} and c_i hold some information in common. To reflect this interaction, or articulation, the u_i is calculated as:

$$u_i = M_1 x_i + M_2 c_i + M_3 u_{i-1} + b \tag{10}$$

Where the matrix $M_3 \in R^{d \times d}$ reflects the interference from u_{i-1}. The intuition behind this recurrent connection is that if the shared neighboring characters have provided much information for u_{i-1}, they should show less effects on u_i.

4 Experiments and results

We evaluate the proposed character-aware NMT model on the Chinese to English translation task. The open-source NMT system, GroundHog [*] (Bahdanau et al., 2014) is used as the baseline system.

4.1 Dataset

For the Chinese to English translation task, the training data consists of 2.3M pairs of sentences. As the traditional RNN search model relies on vector representations for words, we build a fixed vocabulary for each language respectively by choosing 45k of the most frequent words for the source language and 41k of the most frequent words for the target language. Words not included in the vocabulary are replaced with "UNK". For the proposed character-aware NMT model, we build a fixed character vocabulary with the size of 7009 for Chinese, which covers all of the Chinese characters in the training data. The vocabulary for English is the same with the traditional RNN search model. We use the BLEU metric to evaluate the translation quality and test the translation performance on IWSLT04, IWSLT05, IWSLT07, IWSLT08, MT08 and MT12.

4.2 Training detail

In the character-aware NMT model, the character embedding of the source side and the word embedding in the target side are all regarded as part of the model's parameters, and initialized by Gaussian distribution or uniform distribution, same as other parameters of the model. The window size of the row convolution τ is a hyper-parameter which can be set by the user ahead of time. In our implementation, we test the influence of τ by setting it as two, three and four respectively. We use parallel corpus to train RNN search model on a cluster with 8 Tesla K40 GPUs and it takes about 3 days to train the model for a total of 6 epochs. We use the same corpus to train the character-aware NMT model on the same cluster and the training time is longer than RNN search model: 4 days are needed to train the character-aware model for 6 epochs.

4.3 Impacts of the window size

The window size τ is a hyper-parameter which can be set by the user beforehand and it shows great impacts on the translation performance of our proposed model. Table 1 shows the translation performance of the character-aware NMT model when the window size is set as two, three and four respectively.

[*] https://github.com/lisa-groundhog/GroundHog

From table 1, it's easily to be found that the character-aware NMT model achieves the best performance when the window size is set as two. When the window size comes to four, the model can't be trained to converge. We explain this as that when the window size set as four, the neighboring word-level representations share too much information so that there is no distinction between the inputs to the bidirectional RNN. Hence the model may get confused and uneasily to be trained to converge.

Window size	IWSLT04	IWSLT05	IWSLT07	IWSLT08
2	50.01	52.13	33.10	43.15
3	48.62	51.30	31.15	41.26
4	—	—	—	—

Table 1: The impacts of the window size on the character-aware NMT model which has the recurrent connections.

4.4 Results on Chinese-English translation

Table 2 shows the BLEU score on Chinese-English test sets. The word embedding in traditional RNN search model and the character embedding in the proposed character-aware NMT model are all initialized to 512 dimensions by Gaussian distribution. The window size τ is set as two. In table 2, the RNNsearch-Word is the traditional RNN search model which serves as a baseline. To show the ability of our proposed character-aware NMT model, we also test the performance of the model RNNsearch-Char. The only difference between the RNNsearch-Char and the RNNsearch-Word is that the former regards the input sentence as a sequence of characters and the latter regards it as a sequence of words. The Character-aware-forward is the proposed character-aware NMT model which composes the word-level representation with the forward character composing layer and the Character-aware-recurrent utilize the recurrent character composing. By comparing the RNNsearch-word and RNNsearch-Char, we can find that the traditional RNN search model is incapable of handling the case where the input is a sequence of characters. Compared to the RNNsearch-Char, the proposed Character-aware-forward model leads to improvement up to 1.4 BLEU points although its performance is still worse than the baseline of RNNsearch-Word. The Character-aware-recurrent model leads to more significant improvement than the RNNsearch-Char and achieves comparable results with RNNsearch-Word.

Model	IWSLT04	IWSLT05	IWSLT07	IWSLT08	MT08	MT12
RNNsearch-Word	50.14	51.99	33.12	43.02	20.66	20.20
RNNsearch-Char	45.18	49.33	31.29	40.72	17.64	18.39
Character-aware-forward	47.56	50.74	32.48	42.17	19.25	19.65
Character-aware-recurrent	50.01	52.13	33.10	43.15	20.58	20.31

Table 2: The results on the Chinese to English translation tasks.

4.5 Comparison between the summed and concatenated row convolution

To show the effectiveness of our proposed concatenated row convolution, we compare the translation performance between the bidirectional summed row convolution and the bidirectional concatenated row convolution. Both of the two models have recurrent connections in the row convolution layer and the window size are both set as two. From table 3, we can find that the concatenated row convolution outperforms the summed row convolution at every test set. We conjecture that the summed row convolution have lost the information of the context vectors' relative position, which may be vital for composing word-level representation from characters.

Model	IWSLT04	IWSLT05	IWSLT07	IWSLT08
Concatenated row convolution	50.01	52.13	33.10	43.15
Summed row convolution	49.14	50.56	32.48	41.83

Table 3: The comparison between the summed and concatenated row convolution.

RNNsearch_Word	Character-aware-recurrent
Source: 我 的 名字 叫 ***铃木直子*** 。 Translation: My name is <unk>.	Source: 我 的 名 子 叫 ***铃 木 直 子*** 。 Translation: My name is Naoko Suzuki.
Source: 你 知道 ***清水寺*** 在 哪儿 吗？ Translation: Do you know where the <unk> is ?	Source: 你 知 道 ***清 水 寺*** 在 哪 儿 吗？ Translation: Do you know where the water-temple is ?
Source: 你好， ***艾米 哈里斯*** 夫人。 Translation: Hello, MS Amy.	Source: 你 好， ***艾 米 哈 里 斯*** 夫 人 。 Translation: Hello, MS Amy Harris.

Table 4: The translation performance on name entity.

4.6 Performance on name entity and unknown words

To our surprise, the character-aware NMT model is able to translate the name entity very well, as shown in table 4. According to table 4, we can see that the proposed model achieves better translation performance on name entity than the traditional word-based RNN search model. This is partly because that the name entity usually occurs rarely in the training corpus and is often mapped to an unknown word by the RNN search model. However, in character-aware NMT model, the name entity is split into a sequence of characters and each character can be found in the vocabulary. Despite that we still use a word-based decoder in the proposed character-aware NMT model, the number of unknown words in output sentences has decreased dramatically.

5 Conclusions and future work

In this work, we present a novel and simple character-aware NMT model which encodes the input sentence at the character-level. In addition to be applied to the language that has a clear boundary between words, the proposed model is also applied to the language without explicit word segmentation. Hence, no relying on the word boundaries is the most obvious advantage of our model. We firstly introduce the row convolution into NMT and test the effectiveness of several different row convolution methods. Experimental results show that the proposed character-aware NMT model can achieve comparable results with the traditional word-based RNNsearch model. Despite the target side of the proposed model is still word based, the number of unknown words in the output sentences get decreased dramatically. Moreover, the character-aware NMT model shows its superiority on the translation of name entities.

One limitation of our model is that the decoder is still word based. However, this has allowed us a more fine-grained analysis. But in the future, a setting where the target side is also represented as a character sequence must be investigated.

Acknowledgements

This work is supported by National Program on Key Basic Research Project of China(973 Program)(Grant No.2013CB329302). We would like to thank Xu Shuang for her preparing data used in this work. Additionally, we also want to thank Chen Zhineng, Li Jie, Geng Wang, Wang Wenfu and Zhao Yuanyuan for their invaluable discussions on this work.

References

Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al. 2015. Deep speech 2: End-to-end speech recognition in english and mandarin. *arXiv preprint arXiv:1512.02595*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Sébastien Jean Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. *arXiv preprint arXiv:1601.01085*.

David Golub and Xiaodong He. 2016. Character-level question answering with attention. *arXiv preprint arXiv:1604.00727*.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with smt features. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. *EMNLP*, pages 1700–1709.

Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015a. Finding function in form: Compositional character models for open vocabulary word representation. *arXiv preprint arXiv:1508.02096*.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015b. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Minh-Thang Luong and Christopher D Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, pages 2265–2273.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2015. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, pages 3104–3112.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Coverage-based neural machine translation. *arXiv preprint arXiv:1601.04811*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.