A Simple and Fast Strategy for Handling Rare Words in Neural Machine Translation

Minh-Cong Nguyen-Hoang[1], **Thi-Vinh Ngo**[2], **Van-Vinh Nguyen**[1]

[1]***University of Engineering and Technology, VNU, Hanoi, Vietnam***
[2]***University of Information and Communication Technology, TNU, Thai Nguyen, Vietnam***
congnhm@vnu.edu.vn, ntvinh@ictu.edu.vn, vinhnv@vnu.edu.vn

Abstract

Neural Machine Translation (NMT) has currently obtained state-of-the-art in machine translation systems. However, dealing with rare words is still a big challenge in translation systems. The rare words are often translated using a manual dictionary or copied from the source to the target with original words. In this paper, we propose a simple and fast strategy for integrating constraints during the training and decoding process to improve the translation of rare words. The effectiveness of our proposal is demonstrated in both high and low resource translation tasks, including the language pairs: English → Vietnamese, Chinese → Vietnamese, Khmer → Vietnamese, and Lao → Vietnamese. We show the improvements of up to +1.8 BLEU scores over the baseline systems.

1 Introduction

Neural Machine Translation (NMT) (Cho et al., 2014; Sutskever et al., 2014; Vaswani et al., 2017) has recently shown impressive results compared to Statistical Machine Translation (SMT) (Wu et al., 2016; Klein et al., 2017). However, NMT systems still have great challenges (Koehn and Knowles, 2017), in that, addressing rare words is one of them. Due to NMT that has tended to bias in high-frequency words, low-frequency words have little chance of being considered in the inference process. To tackle this problem, some previous works have proposed various strategies to augment translation of low-frequency words. Typically, Luong et al. (2015a) demonstrate the effectiveness of NMT systems by replacing rare words by special symbols such as $unk_1, unk_2, ..., unk_i$ in the sentence. They use an aligned dictionary to map between an unk_i in the source sentence and an other unk_j in the target sentence. This approach tends to raise ambiguity in context of sentence as shown in (Sennrich et al., 2016). In addition, Sennrich et al. (2016) suggest for applying Byte Pair Encoding (BPE) (Gage, 1994) to NMT systems. This technique significantly reduces the vocabulary size and shows substantial improvements on performance translation, and it is widely applied for almost all translation systems nowadays. Following this approach, a rare word will be split into sub-words and the sentence context is still preserved, nevertheless, other new rare words can be also generated. Moreover, this segmentation could make it more difficult to discover original rare words from their sub-words.

To overcome this issue, Vinyals et al. (2015) propose *pointer networks* which automatically copy rare-words from the source sentence into the target sentence. To achieve this aim, they integrate a copy probability to the output distribution with a copy coefficient learnt during the training process. There are also some other variances of these networks such as (Gulcehre et al., 2016; Pham et al., 2018; Song et al., 2019). However, these techniques often copy only a part of rare words when they are separated into sub-words, and we find that they do not have benefit in the data sparsity situation.

Inspired by above *pointer networks*, we propose a simple and fast idea for representing the output probability distribution, though our strategy does not require learning any additional weights. Besides, we leverage neighboring words to identify the suitable position of translation in the source side that corresponds to a rare word in the target side when it has the wrong attention. The proposal is only performed during the inference process, therefore, it does not affect the training. Our experiments show the improvements overcoming the baseline systems.

The background of NMT is shown in 2. The detail of our method is described in section 3, experiments and results are shown in section 4. Finally, the related work is presented in section 5.

Proceedings of the AACL-IJCNLP 2022 Student Research Workshop, pages 40–46
November 20, 2022.

2 Neural machine translation

The goal of NMT systems is to translate a source sentence $x = x_1, ..., x_{|x|}$ to a target sentence $y = y_1, ..., y_{|y|}$. NMT has suggested in (Cho et al., 2014; Sutskever et al., 2014) with recurrent neural networks (RNNs) which use GRU (Gated Recurrent Unit) or LSTM (Long Short Term Memory) for handling the memory context of long sentences. However, these networks face the difficulty of parallel computations during the training process. Vaswani et al. (2017) presents the Transformer model to overcome this issue, which has shown the state-of-the-art in current NMT systems. The probability P (y|x, $\boldsymbol{\theta}$) indicate a NMT model (Vaswani et al., 2017) parameterized by $\boldsymbol{\theta}$. During the training process, parameters are optimized by minimizing the maximum likelihood of the sentence pairs:

$$L(y|x, \boldsymbol{\theta}) = \frac{1}{|y|} \sum_{k=1}^{|y|} logP(y_t|y_{<t}, x, \boldsymbol{\theta}), \quad (1)$$

in there, $y_{<t}$ is a partial translation.

Self-Attention In the Transformer architecture, both Encoder and Decoder are stacks of I identical layers, each layer contains number heads of self-attention to learn context representations.

$$\text{attn}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V, \quad (2)$$

where K (key), Q (query), V (value) are vectors which present the hidden states of tokens in the input sequences and d is the size of hidden states.

Attention mechanism in Transformer between the source sentence and the target sentence is a variance of the attention proposed by Luong et al. (2015b). Its presentation is the same as in 2, however, keys and values are representations of the source sentence while queries are those of the target sentence.

3 Rare words Translation

This section will present details of steps in our method to enhance the translation system.

Tagging data In the first place, we use Giza++[1] to align between source and target sentences. Secondly, a dictionary will be generated from the alignment table. Finally, We tag a special token for rare words in both source and target sentences as in the Figure 1. In our experiments, words that have a frequency below 4 are considered as rare words. Each rare words is inside a pair of "#". We hope that these tokens will help translation systems detect rare words during the training and decoding process.

Inference Inspired by the *pointer network* (Vinyals et al., 2015)

$$P_{output}(y|x, \boldsymbol{\theta}) = \alpha * P_g + (1 - \alpha) * P_c, \quad (3)$$

where, α is a copy factor which is learnt during the training, P_g is the normal output distribution of the model while P_c is the copy distribution which presents the target-to-source attention weights to indicate which tokens will be copied from the source sentences to the target sentence. There are also some variants of P_c in previous works (Vinyals et al., 2015; Gulcehre et al., 2016; Song et al., 2019; Pham et al., 2018). However, because the copy factor α is automatically learnt in the training, therefore, it may be not good enough in data sparsity situations to perform the given aim, and this may lead to wrong predictions for both rare words and non rare words.

To address this issue, we fix the copy factor α to constants during the inference process. Base on tagged labels as mentioned above to the detection rare words, we set α to 1 after detecting the token *"#"* that mark the start position of each rare word in the step i^{th}, otherwise, it is set to 0 when the other stop token *"#"* is discovered.

Heuristic Due to the systems that could attend to the translation of a rare word in the target side to the wrong translation in the source side may contain many rare words. In this issue, around words are also used to support the inference process discovering the best suitable position of the corresponding translation in the source sentence. To implement this idea, we define a heuristic function $score_{ij}$ to estimate alignment weight between each rare word i^{th} in the target side and the rare word j^{th} in the source side during the inference process as in the formula 4.

$$score_{ij} = \beta * \sum P_{sw_{kj}} + \gamma * \sum P_{\#_{mj}} + \epsilon * \sum P_{a_{lj}} \quad (4)$$

where $P_{sw_{kj}}$ is the attention weight between the sub-word k^{th} of the rare word i^{th} in the target side and the rare word j^{th} in the source side. Similarly, $P_{\#_{mj}}$ is the attention weight of the token *"#"* m^{th}

[1] https://github.com/moses-smt/giza-pp

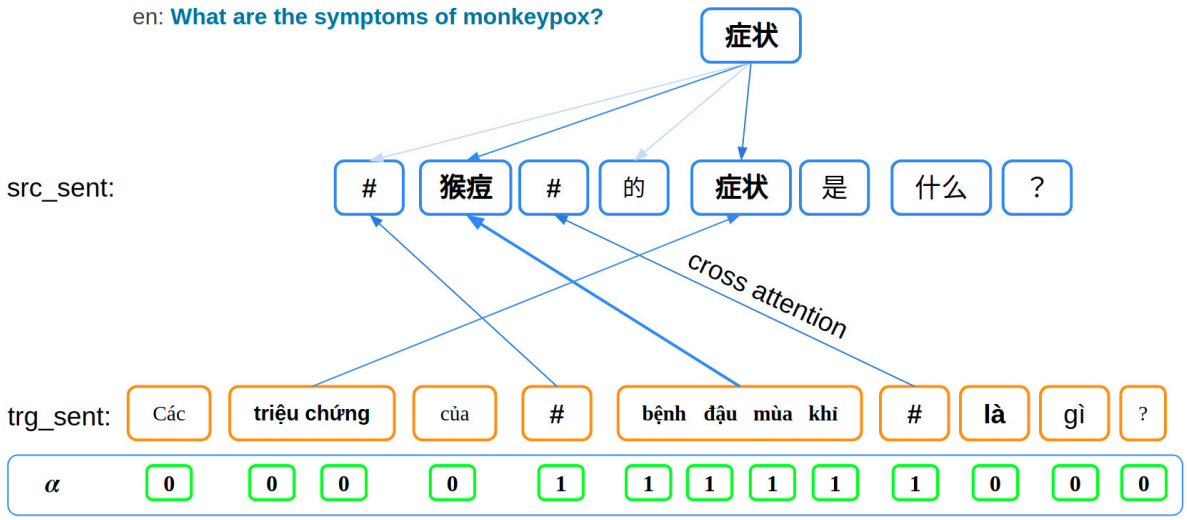

Figure 1: A illustration of our method: the rare word "bệnh đậu mùa khỉ (猴痘)" is put on a pair of *"#"*, the non rare word "triệu trứng" is used to support the inference process detecting the suitable position of translation in the source side. The α is a copy factor which is described in the formula 3

.

that starts or ends the rare word i^{th}, and $P_{a_{lj}}$ is the attention weight of the around word l^{th} of the rare word i^{th}. In the experiment, we only consider two content words that are nearest to the rare word i^{th}, in which, one word before and another one after the the rare word i. $P_{sw_{kj}}$, $P_{\#_{mj}}$, and $P_{a_{lj}}$ will assigned to 1 if they attend to the rare word j^{th} in the source side, otherwise, they are 0. Besides, β, γ and ϵ are constants, in our experiments, both β and γ are the same value and they are assigned to 0.4 while ϵ is 0.2.

The rare word j^{th} in the source side that has the highest score corresponding to the rare word i^{th} in the target side will be chosen as the its best translation. Our algorithm is detailed in the Algorithm 1.

Thus, at each step n^{th}, the cross attention weights will be computed for each target token. We utilize head 0 with layer 0 in order to evaluate these attentions .To save time, we employ *"Cache Maintenance"* strategy inspired by (Yan et al., 2021) to archive Q_t, K_t, V_t, and $attn_t$ in beam search.

4 Experiments

This section presents our implementation of the translation systems. Our method show the efficiency of both bilingual and multilingual translation systems. The SacreBLEU score (Post, 2018)[2] is employed to evaluate the quality translation.

[2] https://github.com/mjpost/sacrebleu

4.1 Datasets and Training System

Datasets and Pre-processing We use different datasets from KC4.0 UET (Nguyen et al., 2022), Hugging Face[3], and Asian Language Treebank[4] (ALT) corpus. For all parallel corpus collected from Hugging Face, we filter and remove poor quality sentence pairs using LASER[5].

For all experiments, the development and test sets from the Asian Language Treebank (ALT) corpus are utilized for early stop and evaluate the efficiency of our strategy. The development set includes 1000 sentence pairs while the test set contains 1018 other ones.

To generate alignment dictionaries for tagging rare words, we use various segmentation tool for each language: pkuseg[6] for Chinese texts, laoNLP[7] for Laos texts, khmernltk[8] for Khmer texts, and moses[9] tokenizer for English and Vietnamese texts.

[3] https://huggingface.co

[4] https://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/

[5] https://github.com/facebookresearch/LASER

[6] https://github.com/lancopku/pkuseg-python

[7] https://github.com/wannaphong/LaoNLP

[8] https://github.com/VietHoang1512/khmer-nltk

[9] https://github.com/moses-smt/mosesdecoder/tree/master/scripts

Algorithm 1: Finding the suitable position in the source sentence for a target rare word

Input :
$attn\_cache$ contains cross attentions,
$marked\_src\_sent$: marks the position of the rare word,
$word\_query$: the translation of the rare word in target sentence including special token "#".
Output : The best candidate correspond to rare word

$P_{sw_{kj}}, P_{\#_{mj}}, P_{a_{lj}} = \{0\}$ `// Initial with 0 and their size is number of the rare words.`
p_words = $spm(word\_query)$ `// the number of sub words of word query.`
`/* Compute` $P_{sw_{kj}}$ `: */`
for $i \leftarrow 0$ **to** $n - 1$ **do**
  piece = p_words[i]
  attn_pos = arg_max(attn_cache[piece])
  rare_word = marked_src_sent[attn_pos]
  if *rare_word is not equal 0* **then**
    $P_{sw_{kj}}$[rare_word-1] += 1
  end
end
`/* Compute` $P_{\#_{mj}}$ `: */`
for $i \leftarrow 0$ **to** *2* **do**
  `// Two "#" nearest the rare word`
  attn_pos = arg_max(attn_cache[$\#_i$])
  rare_word = marked_src_sent[attn_pos]
  if *rare_word is not equal 0* **then**
    $P_{\#_{mj}}$[rare_word-1] += 1
  end
end
`/* Compute` $P_{a_{lj}}$ `: */`
neighbor_word = get_neighbor(word_query)
`// get two nearest neighbor words`
while $neighbor\_word \neq None$ **do**
  pos = attn_cache[neighbor_word$_i$]
  attn_pos = arg_max(pos)
  rare_word = find_nearest(attn_pos)
  `// Find the nearest rare word from attention position`
  if *rare_word is not equal 0* **then**
    $P_{a_{lj}}$[rare_word-1] += 1
  end
end
$score_{ij} = \beta * P_{sw_{kj}} + \gamma * P_{\#_{mj}} + \epsilon * P_{a_{lj}}$
rare_word = arg_max($score_{ij}$)
return $rare\_word$

No.	Lang	Size	Source
1	**Zh-Vi**	500k	KC4.0 UET
		2M	HuggingFace
		18k	ALT
2	**Lo-Vi**	150k	KC4.0 UET
		18k	ALT
3	**Km-Vi**	150k	KC4.0 UET
		18k	ALT
4	**En-Vi**	2.6M	HuggingFace

Table 1: The statistics of parallel datasets are used in our experiments.

For bilingual systems, we apply sentencePiece[10] (Kudo and Richardson, 2018) with split-digit option and 32K joint merge operations for the original texts in all languages. We estimate our proposed in the Chinese → Vietnamese pair including 150K sentence pairs which extracted from KC4.0 UET corpus in the low-resource issue and 2.5M which are concatenated from Hugging Face and KC4.0 UET corpus in the higher resource situation.

For the multilingual system, we mix all the parallel corpus as described in table 1 and gain approximately 5M5 sentence pairs. The texts from the baseline systems are tagged for rare words as described in the section 3.

System and training We implement our baseline NMT systems using the framework ViNMT[11] (Quan et al., 2021). All settings are the same for both bilingual and multilingual systems. The training system includes 6 layers for both encoder and decoder, the sizes of hidden states and embedding is 512, the number of heads are 8. The Adam Optimizer is used to optimize parameters of the whole model with the initial learning rate is 1e-3. The size of each mini-batch is 64 sentence pairs. The other settings are the defaults of ViNMT.

To apply our ideas to the NMT system, we modify the baseline architecture following the steps in the section 3. Besides, the baseline architecture is also reformed as in Song et al. (2019) for comparison purpose.

All systems are trained until they gain convergence on the development set. The best model in terms of unigram accuracy on the validation set is used to translate the test set with beam size of 4.

[10] https://github.com/google/sentencepiece
[11] https://github.com/KCDichDaNgu/KC4.0_MultilingualNMT

4.2 Results

The practical results are presented in Table 2 and Table 3.

Bilingual systems Our baseline systems have achieved 18.1 and 28.1 BLEU scores on two datasets. To estimate the efficiency of our method, the other strategy for dealing with the rare words in Song et al. (2019) are performed in our experiments. Our proposal overcomes both the baseline system and Song's system Song et al. (2019), and they have gained improvements of +0.2 and +1.0 BLEU scores in both two datasets.

No.	Systems	150K	2.5M
1	Baseline	18.1	28.1
3	Song et al. (2019)	17.5	28.0
4	Our proposal	**18.3**	**29.1**

Table 2: BLEU scores on ALT test set for bilingual systems for the Chinese → Vietnamese translation task when applying our method.

Multilingual system To further investigate the efficiency of our proposal, we train a multilingual system "many to one" from English (En), Chinese (Zh), Laos (Lo), and Khmer (Km) to Vietnamese. The result is shown in the Table 3.

No.	Lang	Baseline	Our proposal
1	En-Vi	32.4	**34.0** (+1.6)
2	Zh-Vi	28.0	**29.8** (+1.8)
3	Lo-Vi	24.4	**25.1** (+0.7)
4	Km-Vi	28.9	**29.2** (+0.3)

Table 3: The BLEU scores for the multilingual translation system

Our method has achieved significant improvements on almost translation tasks. In particular, it gains +1.6 BLEU scores for English → Vietnamese translation, and + 1.8 BLEU points for Chinese → Vietnamese translation. The translation task Laos → Vietnamese obtains +0.7 BLEU scores while it only acquires +0.3 BLEU points for the Khmer → Vietnamese translation task.

5 Related Work

Dealing with rare words has investigated by many previous works in machine translation. Tsvetkov and Dyer (2015) employed a model of lexical borrowing to enhance SMT systems. However, this approach claim extraction of complex features such as phonetic and semantic features, or pre-trained SMT systems, and it is difficult to apply to NMT systems. Jean et al. (2015) used a large vocabulary to solve the rare words but this increases parameters and leads to augmentation the size of models and rare words still exist. Some other works require additional resources to tackle rare words such as Trieu (2016) exploited word similarity, or Ngo et al. (2019) utilize synonyms to advance translation systems. Luong et al. (2015a) employed special symbols to present rare words or unknown words but this tend to increase the ambiguous of sentence context. Sennrich et al. (2016) applied BPE algorithm to separate rare words into sub-words, however, new sub-words are again generated. Furthermore, Vinyals et al. (2015) suggested point networks allow to copy automatically rare words from source side to the target side with learning supplemental parameters, and in some case, it has only a part of rare words (sub-words) are copied. This approach is also considered in recent studies Gulcehre et al. (2016); Song et al. (2019); Pham et al. (2018).

Our proposal also relies on the idea of point networks, nevertheless, we fix the copy factor to a constant. Besides, we leverage neighbouring words to specify the best position of the rare word in the source side that corresponds to the one in the target side.

6 Conclusion

In this work, we propose a simple and fast method to improve the translation quality for rare words. Our technique does not require training supplemental parameters, and this strategy is only performed in the inference process, therefore, the training time does not change. In the future, we would like to consider more neighbouring words around rare words for selecting position to improve the quality of translation tasks.

Acknowledgments

This work is supported by Ministry of Science and Technology of Vietnam under Program KC 4.0, No. KC-4.0.12/19-25.

References

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. In *Proceedings*

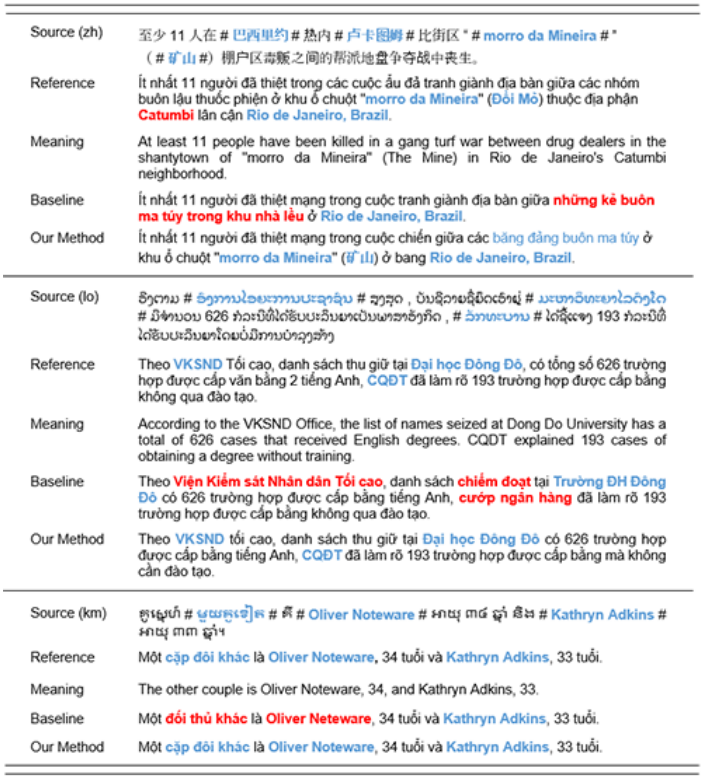

Figure 2: Examples of outputs from our multilingual translation systems with the proposed methods compare to the baseline systems for Zh → Vi, Lo → Vi, and Km → Vi translation tasks.

of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. In *C Users J., 12(2):23–38, February*.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words.

S'ebastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1–10. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Minh-Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015a. Addressing the rare word problem in neural machine translation.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Thi-Vinh Ngo, Thanh-Le Ha, Phuong-Thai Nguyen, and Le-Minh Nguyen. 2019. Overcoming the rare word problem for low-resource language pairs in neural machine translation. *Proceedings of the 6th Workshop on Asian Translation*.

Van-Vinh Nguyen, Ha Nguyen-Tien, Huong Le-Thanh, Phuong-Thai Nguyen, Van-Tan Bui, Nghia-Luan Pham, Tuan-Anh Phan, Minh-Cong Nguyen Hoang, Hong-Viet Tran, and Huu-Anh Tran. 2022. Kc4mt: A high-quality corpus for multilingual machine translation.

Ngoc-Quan Pham, Jan Niehues, and Alexander Waibel. 2018. Towards one-shot learning for rare-word translation with external experts. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 100–109, Melbourne, Australia. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on*

Machine Translation: Research Papers, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Nguyen Hoang Quan, Nguyen Thanh Dat, Minh Cong Nguyen Hoang, Nguyen Van Vinh, Ngo Thi Vinh, Nguyen Phuong Thai, and Tran Hong Viet. 2021. Vinmt: Neural machine translation tookit. *CoRR*, abs/2112.15272.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Nguyen L. M. Nguyen P. T. Trieu, H. L. 2016. Dealing with out-of-vocabulary problem in sentence alignment using word similarity. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Oral Papers (pp. 259-266)*.

Yulia Tsvetkov and Chris Dyer. 2015. Lexicon stratification for translating out-of-vocabulary words. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 125–131, Beijing, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Yu Yan, Fei Hu, Jiusheng Chen, Nikhil Bhendawade, Ting Ye, Yeyun Gong, Nan Duan, Desheng Cui, Bingyu Chi, and Ruofei Zhang. 2021. FastSeq: Make sequence generation faster. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 218–226, Online. Association for Computational Linguistics.