Bridging the Gap between Training and Inference for Neural Machine Translation*

Wen Zhang[3] , **Yang Feng**[1,2†] and **Qun Liu**[4]
[1] Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Smart Platform Product Department of Tencent Inc., China
[4] Huawei Noah's Ark Lab, Hong Kong, China
fengyang@ict.ac.cn, kevinwzhang@tencent.com, qun.liu@huawei.com

Abstract

Neural Machine Translation (NMT) generates target words sequentially in the way of predicting the next word conditioned on the context words. At training time, it predicts with the ground truth words as context while at inference it has to generate the entire sequence from scratch. This discrepancy of the fed context leads to error accumulation among the translation. Furthermore, word-level training requires strict matching between the generated sequence and the ground truth sequence which leads to overcorrection over different but reasonable translations. In this paper, we address these issues by sampling context words not only from the ground truth sequence but also from the predicted sequence during training[1]. Experimental results on NIST Chinese→English and WMT2014 English→German translation tasks demonstrate that our method can achieve significant improvements on multiple data sets compared to strong baselines.

1 Introduction

Neural Machine Translation has shown promising results and drawn more attention recently. Most NMT models fit in the encoder-decoder framework, including the RNN-based [Sutskever *et al.*, 2014; Bahdanau *et al.*, 2015; Meng and Zhang, 2019], the CNN-based [Gehring *et al.*, 2017] and the attention-based [Vaswani *et al.*, 2017] models, which predict next word conditioned on the previous context words, deriving a language model over target words. The scenario is at training time the ground truth words are used as context while at inference the entire sequence is generated by the resulting model on its own and hence the previous words generated by the model are fed as context. As a result, the words at training and inference are predicted from different distributions, namely, from the data distribution as opposed to the model distribution. This discrepancy, called *exposure bias* [Ranzato *et al.*, 2015], leads to a gap between training and inference. As the target sequence grows, the errors accumulate among the sequence and the model has to predict under the condition it has never met at training time.

Intuitively, to relieve this problem, the model should be trained under the same condition it will face at inference. Inspired by DATA AS DEMONSTRATOR (DAD) [Venkatraman *et al.*, 2015], feeding as context both ground truth words and the predicted words during training can be a solution. NMT models usually optimize the cross-entropy loss which requires a strict pairwise matching at the word level between the predicted sequence and the ground truth sequence. Once the model generates a word deviating from the ground truth word, the cross-entropy loss will correct the error immediately and draw the remaining generation back to the ground truth sequence. However, this causes a new problem.

A sentence usually has multiple reasonable translations and it cannot be said that the model makes a mistake even if it generates a word different from the ground truth word. In the training set, for example, there is a sentence pair:

train-src:	wǒ men yīng gāi zūn shǒu guī zé .
train-ref:	We should comply with the rule .

When the source sentence "wǒ men yīng gāi zūn shǒu guī zé ." is fed, Maximum Likelihood Estimation (MLE) adjusts model parameters to generate "We should comply with the rule ." with a strict word-level matching. While we assume the following source sentence as input at inference:

test-src:	wǒ men yīng gāi zūn shǒu fǎ lǜ .
test-ref:	We should abide by the law .
cand1:	We should abide with the rule .
cand2:	We should abide by the law .

once the model generates "abide" as the third target word, since the model **has never seen the pattern "We should abide ..." in the training set**, it would generate "with" as the fourth word (as *cand1*) so as to produce larger sentence-level likelihood, although "by" is the right choice. The translation *cand1* can be treated as *overcorrection* phenomenon. Different from the original method, when the training proceeds to the third step, assuming that the oracle word sampled by our method is exactly "abide", then the model will potentially

[†] Corresponding author. This work was done when Wen Zhang was a student in ICT/CAS.

[1] Our code is available at https://github.com/ictnlp/OR-NMT

[*] This work was originally published in ACL2019. [Zhang *et al.*, 2019]

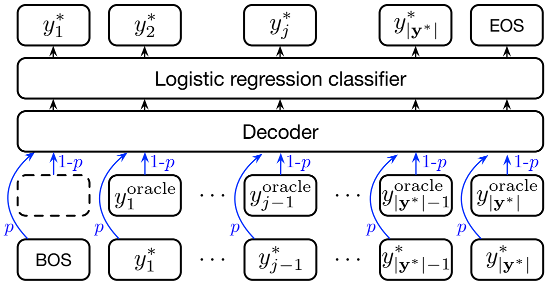

Figure 1: The architecture of our method.

have seen the pattern "We should abide ..." in the training set, which finally outputs the correct translation *cand2*. We refer to this solution as ***O**vercorrection **R**ecovery* (**OR**).

In this paper, we present a method to bridge the gap between training and inference and improve the overcorrection recovery capability of NMT. Our method first selects *oracle* words from its predicted words and then samples as context from the oracle words and ground truth words. Meanwhile, the oracle words are selected not only with a word-by-word greedy search but also with a sentence-level evaluation, e.g. BLEU, which allows greater flexibility under the pairwise matching restriction of cross-entropy. At the beginning of training, the model selects as context ground truth words with a higher probability. As the model converges gradually, oracle words are chosen as context more and more frequently. In other words, the training process shifts from fully-guided scheme to a less-guided scheme, leading the model to have the chance to learn to handle the mistakes made at inference and also has the ability to recover from overcorrection over alternative translations. We verify our approach on both the RNNsearch model and the stronger Transformer model. The results show that our method can significantly improve the performance on both models.

2 The Proposed Method

Suppose a sentence pair consists of the source sentence $\mathbf{x}$ and the target sentence $\mathbf{y}^*$, which are denoted as:

$$\mathbf{x} = \left\{x_1, \cdots, x_{|\mathbf{x}|}\right\}; \ \ \mathbf{y}^* = \left\{y_1^*, \cdots, y_{|\mathbf{y}^*|}^*\right\} \tag{1}$$

As shown in Figure 1, either the ground truth word or the previous predicted word, i.e. *oracle words*, is fed into decoder as context, with a certain probability. This potentially can reduce the gap between training and inference by training the model to handle the situation which will appear during test time. We propose two strategies for selecting oracle words:

- select oracle words at the word level with greedy search
- select a oracle sequence at the sentence-level optimum

The sentence-level oracle provides an option of n-gram matching with the ground truth sequence and hence inherently has the ability of recovering from overcorrection for the alternative context. To predict the j-th target word y_j, the following steps are involved in our approach:

1. Select an oracle word y_{j-1}^{oracle} (at word or sentence level) at the $\{j-1\}$-th step. (Section 2.1)
2. Sample as the context word from the ground truth word y_{j-1}^* with a probability of p or from the oracle word y_{j-1}^{oracle} with a probability of 1-p. (Section 2.2)

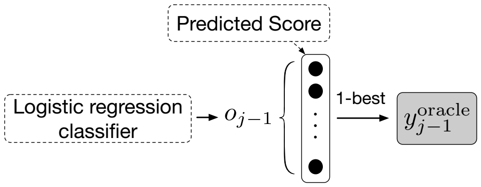

Figure 2: Word-level oracle without noise.

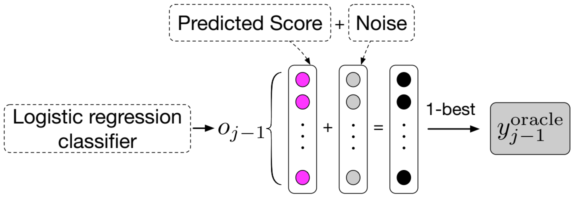

Figure 3: Word-level oracle with Gumbel noise.

2.1 Oracle Word Selection

Generally, at the j-th step, the NMT model needs the ground truth word y_{j-1}^* as the context word to predict y_j, alternatively, we could select an oracle word y_{j-1}^{oracle} to simulate the context word. y_{j-1}^{oracle} should be a word similar to the ground truth or a synonym. Different strategies will produce different oracle words. One option is to employ word-level greedy search to output the oracle word at each step, which is called *Word-level Oracle* (called WO). Besides, we can further optimize the oracle by enlarging the search space with beam search and then re-ranking the candidate translations with a sentence-level metric, e.g. BLEU [Papineni *et al.*, 2002], GLEU [Wu *et al.*, 2016], ROUGE [Lin, 2004], etc. The selected translation is called *oracle sentence*, the words in the translation are *Sentence-level Oracle* (denoted as SO).

Word-Level Oracle. For the j-th decoding step, we assume the decoder predict the final distribution by following equation:

$$\mathcal{D}_j = \mathrm{softmax}\left(o_j\right) \tag{2}$$

where o_j is a vector mapped to the vocabulary, named *predicted score*. The intuitive approach to select the word-level oracle is to pick the word with the highest probability from the word distribution $\mathcal{D}_{j-1}$ drawn by Eq. (2), which is shown in Figure 2. In practice, we can acquire more robust word-level oracles by introducing the *Gumbel-Max* technique [Gumbel, 1954; Maddison *et al.*, 2014], which provides a simple and efficient way to sample from a categorical distribution. The Gumbel noise, treated as a form of regularization, is added to o_{j-1} (as the Figure 3 shows):

$$\eta = -\log\left(-\log u\right) \tag{3}$$
$$\tilde{o}_{j-1} = \left(o_{j-1} + \eta\right)/\tau \tag{4}$$

where η is the Gumbel noise calculated from a uniform random variable $u \sim \mathcal{U}(0,1)$, τ is temperature. As τ approaches 0, the $\mathrm{softmax}$ function is similar to the argmax operation, and it becomes uniform distribution gradually when $\tau \to \infty$. Similarly, according to $\tilde{o}_{j-1}$, the 1-best word is selected as the word-level oracle word[1]:

$$y_{j-1}^{\text{oracle}} = y_{j-1}^{\text{WO}} = \text{argmax}\left(\tilde{o}_{j-1}\right) \tag{5}$$

Note that the Gumbel noise is just used to select the oracle word and it does not affect the loss function for training.

Sentence-Level Oracle. The sentence-level oracle is employed to allow for more flexible translation with n-gram matching required by a sentence-level metric. In this paper, we employ BLEU as the sentence-level metric. To select the sentence-level oracles, we first perform beam search for all sentences in each batch, assuming beam size is k, and get k-best candidate translations. In the process of beam search, we also could apply the Gumbel noise for each word generation. We then evaluate each translation by calculating its BLEU score with the ground truth sequence, and use the translation with the highest BLEU score as the *oracle sentence*. We denote it as $\mathbf{y}^{\text{S}} = (y_1^{\text{S}}, ..., y_{|\mathbf{y}^{\text{S}}|}^{\text{S}})$, then at the j-th decoding step, we define the sentence-level oracle word as

$$y_{j-1}^{\text{oracle}} = y_{j-1}^{\text{SO}} = y_{j-1}^{\text{S}} \tag{6}$$

But a problem comes with sentence-level oracle. As the model samples from ground truth word and the sentence-level oracle word at each step, the two sequences should have the same number of words. However we can not assure this with the naive beam search decoding algorithm. Based on the above problem, we introduce *force decoding* to make sure the two sequences have the same length.

Assume that the length of the ground truth sequence is $|\mathbf{y}^*|$, the goal of force decoding is to generate a sequence with $|\mathbf{y}^*|$ words followed by a special end-of-sentence (EOS) symbol. Therefore, in beam search, once a candidate translation tends to end with EOS when it is shorter or longer than $|\mathbf{y}^*|$, we will force it to generate $|\mathbf{y}^*|$ words, that is,

- If the candidate translation gets a word distribution $\mathcal{D}_j$ at the j-th step where $j \leqslant |\mathbf{y}^*|$ and EOS is the top first word in $\mathcal{D}_j$, then we select the top second word in $\mathcal{D}_j$ as the j-th word of this candidate translation.
- If the candidate translation gets a word distribution $\mathcal{D}_{|\mathbf{y}^*|+1}$ at the $\{|\mathbf{y}^*|+1\}$-th step where EOS is not the top first word in $\mathcal{D}_{|\mathbf{y}^*|+1}$, then we select EOS as the $\{|\mathbf{y}^*|+1\}$-th word of this candidate translation.

In this way, we can make sure that all the k candidate translations have $|\mathbf{y}^*|$ words, then re-rank the k candidates according to BLEU score and select the top first as the oracle sentence. For adding Gumbel noise into the sentence-level oracle selection, we replace the $\mathcal{D}_j$ with $\tilde{\mathcal{D}}_j$ at the j-th decoding step during force decoding.

2.2 Sampling with Decay

In our method, we employ a sampling mechanism to randomly select the ground truth word y_{j-1}^* or the oracle word y_{j-1}^{oracle} as y_{j-1}. At the beginning of training, as the model is not well trained, using y_{j-1}^{oracle} as y_{j-1} too often would lead to very slow convergence, even being trapped into local optimum. On the other hand, at the end of training, if the context y_{j-1} is still selected from the ground truth word y_{j-1}^* at a large probability, the model is not fully exposed to the circumstance which it has to confront at inference and hence can not know how to act in the situation at inference. In this sense, the probability p of selecting from the ground truth word can not be fixed, but has to decrease progressively as the training advances. At the beginning, p=1, which means the model is trained entirely based on the ground truth words. As the model converges gradually, the model selects from the oracle words more often.

Borrowing ideas from but being different from Bengio [2015] which used a schedule to decrease p as a function of the index of mini-batch, we define p with a decay function dependent on the index of training epochs e (starting from 0)

$$p = \frac{\mu}{\mu + \exp\left(e/\mu\right)} \tag{7}$$

where μ is a hyper-parameter. The function is strictly monotone decreasing. As the training proceeds, the probability p of feeding ground truth words decreases gradually.

3 Experiments

We conduct experiments on the NIST Chinese→English (Zh→En) and the WMT2014 English→German (En→De) translation tasks.

3.1 Settings

For Zh→En, the training set consists of 1.25M sentence pairs extracted from LDC corpora. We choose the NIST 2002 (MT02) as the validation set, and the NIST 2003~2006 (MT03~06) as the test sets. For En→De, The training set contains 4.5M sentence pairs provided by WMT2014. We use the newstest2013 and newstest2014 as the validation and test set respectively. Byte pair encoding (BPE) [Sennrich *et al.*, 2016] is employed to produce a shared vocabulary of 30k and 37k tokens for Zh→En and En→De. BLEU score [Papineni *et al.*, 2002] is used to evaluate the quality of translation[2]. Besides, we make statistical significance test according to Collins [2005].

3.2 Systems

The following systems are involved:
RNNsearch: An improved version of Bahdanau [2015]
Transformer: Base model[3] [Vaswani *et al.*, 2017] **SS-NMT:** Scheduled sampling (SS) with inverse sigmoid decay [Bengio *et al.*, 2015] based on RNNsearch
MIXER: Sentence-level training with mixed incremental cross-entropy reinforce [Ranzato *et al.*, 2015], where the sentence-level metric is BLEU and the average reward is acquired by its offline method with a 1-layer linear regressor.

[1] In order to simplify the calculation, we do not use softmax, because the softmax operation does not affect the sorting

[2] For Zh→En, case-insensitive BLEU score is calculated by the *mteval-v11b.pl* script. For En→De, we use the *multi-bleu.pl* script to calculate case-sensitive tokenized BLEU score.

[3] https://github.com/pytorch/fairseq

		Zh→En					*En→De*
Systems	**Architecture**	**MT03**	**MT04**	**MT05**	**MT06**	**Average**	**newstest2014**
Existing end-to-end NMT systems							
Tu [2016]	Coverage	33.69	38.05	35.01	34.83	35.40	–
Shen [2016]	MRT	37.41	39.87	37.45	36.80	37.88	–
Zhang [2017]	Distortion	37.93	40.40	36.81	35.77	37.73	–
Our end-to-end NMT systems							
this work	RNNsearch	37.93	40.53	36.65	35.80	37.73	25.82
	+ SS-NMT	38.82	41.68	37.28	37.98	38.94	26.50
	+ MIXER	38.70	40.81	37.59	38.38	38.87	26.76
	+ OR-NMT	**40.40**‡†*	**42.63**‡†*	**38.87**‡†*	**38.44**‡	**40.09**	**27.41**‡
	Transformer	46.89	47.88	47.40	46.66	47.21	27.34
	+ OR-NMT	**48.31***	**49.40***	**48.72***	**48.45***	**48.72**	**28.65**‡

Table 1: Case-insensitive BLEU scores (%) on Zh→En translation task. "‡", "†", "⋆" and "∗" indicate statistically significant difference (p<0.01) from RNNsearch, SS-NMT, MIXER and Transformer, respectively.

OR-NMT: Our proposed model. For the sentence-level oracle selection, we set the beam size to be 3, set τ=0.5 in Eq. (4) and μ=12 for the decay function in Eq. (7). OR-NMT is the abbreviation of NMT with Overcorrection Recovery.

3.3 Results on Zh→En and En→De Translation

We verify our method on two baseline models with the NIST Zh→En and WMT2014 En→De datasets.

Results on NIST Zh→En Translation Task

As shown in Table 1, Compared with the three existing models, our RNNsearch baseline 1) outperforms previous shallow RNN-based NMT system equipped with the coverage model; and 2) achieves competitive performance with the MRT and the Distortion on the same datasets. We hope that our shallow RNNsearch baseline makes the evaluation convincing.

From the Table 1, we can see that both the scheduled sampling and MIXER have brought a certain improvement to the baseline system by mitigating the exposure bias problem. Compared with them, OR-NMT further brings a significant improvement by about +1.2 BLEU points averagely on four test sets. It is worth noting that OR-NMT averagely outperforms the RNNsearch and Transformer[4] baseline systems by 2.4 and 1.5 BLEU points, respectively.

Results on WMT2014 En→De Translation Task. We also evaluate our approach on the WMT2014 En→De translation task. Similar to the results on the Zh→En translation task, we can see from Table 1 that the proposed method can outperforms the related approaches on RNNsearch baseline, and when our method is applied to the RNNsearch and Transformer baseline systems, translation performance can be improved by +1.6 and +1.3 BLEU points, respectively. The results prove that our method works well across different language pairs.

Systems	Average
RNNsearch	37.73
+ word-level oracle	38.94
+ noise	39.50
+ sentence-level oracle	39.56
+ noise	**40.09**

Table 2: Factor analysis on Zh→En translation, the results are average BLEU scores on MT03∼06 datasets.

3.4 Factor Analysis

We propose three strategies to mitigate the overcorrection problem, including word-level oracle, sentence-level oracle, and Gumbel noise. To explore the influence of these factors, we conduct ablation experiment and list the results in Table 2.

When employing only the word-level oracle, the translation performance was improved by +1.21 BLEU points, this indicates that feeding predicted words as context can mitigate exposure bias. Sentence-level oracle can further achieve +0.62 BLEU points improvement. It shows that the sentence-level oracle performs better than the word-level oracle in terms of BLEU. We conjecture that the superiority may come from a greater flexibility for word generation which can mitigate the problem of overcorrection. By incorporating the Gumbel noise during the generation of the word-level and sentence-level oracle words, the BLEU score are further improved by 0.56 and 0.53 respectively. This indicates Gumbel noise is helpful for sampling oracle words, which is consistent with our claim that Gumbel-Max provides a efficient and robust way to sample from a categorical distribution.

4 Conclusion

We proposed a method to bridge the gap between training and inference for NMT. Experimental results show that our method can produce significant improvement and increase the ability of the model to recover from overcorrection.

Acknowledgments

This work was supported by National Key R&D Program of China (NO.2017YFE0192900 and NO. 2018AAA0102500).

[4] To avoid breaking the parallelism of the training, word-level oracle is obtained through greedy search, which is the same as the case where beam size is set to 1 when the sentence-level oracle is sampled.

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR 2015*, 2015.

[Bengio *et al.*, 2015] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc., 2015.

[Collins *et al.*, 2005] Michael Collins, Philipp Koehn, and Ivona Kucerova. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[Gehring *et al.*, 2017] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[Gumbel, 1954] Emil Julius Gumbel. Statistical theory of extreme valuse and some practical applications. *Nat. Bur. Standards Appl. Math. Ser. 33*, 1954.

[Lin, 2004] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[Maddison *et al.*, 2014] Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3086–3094. Curran Associates, Inc., 2014.

[Meng and Zhang, 2019] Fandong Meng and Jinchao Zhang. Dtmt: A novel deep transition architecture for neural machine translation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI'19. AAAI Press, 2019.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

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

[Sennrich *et al.*, 2016] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[Shen *et al.*, 2016] Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1683–1692, 2016.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[Tu *et al.*, 2016] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August 2016. Association for Computational Linguistics.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[Venkatraman *et al.*, 2015] Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3024–3030. AAAI Press, 2015.

[Wu *et al.*, 2016] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[Zhang *et al.*, 2017] Jinchao Zhang, Mingxuan Wang, Qun Liu, and Jie Zhou. Incorporating word reordering knowledge into attention-based neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1524–1534, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[Zhang *et al.*, 2019] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy, July 2019. Association for Computational Linguistics.