Neural Machine Translation with Bilingual History Involved Attention

Haiyang Xue[1], Yang Feng[1], Di You[2], Wen Zhang[1], and Jingyu Li[1]

[1] Key Laboratory of Intelligent Information Processing Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
xuehaiyang,fengyang,zhangwen,lijingyu@ict.ac.cn
[2] Worcester Polytechnic Institute, Worcester, MA, USA
dyou@wpi.edu

Abstract. The using of attention in neural machine translation (NMT) has greatly improved translation performance, but NMT models usually calculate attention vectors independently at different time steps and consequently suffer from over-translation and under-translation. To mitigate the problem, in this paper we propose a method to consider the translated source and target information up to now related to each source word when calculating attentions. The main idea is to keep track of the translated source and target information assigned to each source word at each time step and then accumulate these information to get the completion degree for each source word. In this way, in the later calculation of the attention, the model can adjust the attention weights to give a reasonable final completion degree for each source word. Experimental results show that our method can outperform the strong baseline systems significantly both on the Chinese-English and English-German translation tasks and produce better alignment on the human aligned data set.

Keywords: Neural Machine Translation · Bilingual History Information · Attention Mechanism

1 Introduction

Neural machine translation (NMT) [1–3, 12, 15] has made great progress and drawn much attention recently. NMT models mainly fit in the attention-based encoder-decoder framework where the encoder encodes the source sentence into representations in a common semantic space and at each time step the decoder first collects source information over all the source words via an attention function and then generates a target word based on the collected source information.

Although there may exist different attention functions, including additive attention and dot-product attention [15], the main mechanism is almost the same which first gets the weight for each source representation according to its relevance to the current target-side information and then outputs the weighted sum of source representations as the source information for each time step to translate. From this process, we can see that the calculation of the attention at each time step is only related to the current target-side information and the keys (usually the representations of source words). It does not

involve the previous attention directly and hence is independent to each other at different time steps. As a result, the attention component cannot get to know the completion degree of each source word which leads to *over-translation* or *under-translation* [13]. Table 1 gives examples of over-translation and under-translation. Example (1) shows the case of over-translation where "23" has been translated twice. If the model can get the translation derived from "23", it may not attend too much on it when calculating attention. Example (2) indicates the case of under-translation where the source words "5 zhōunián" have not been translated. Once the model can get the translated part of "5 zhōunián", it will adjust to give more attention to it. As a conclusion, if the model can maintain the translated source and target translation up to now related to each source word, it can work out more reasonable attention. On these grounds, in order to address

(1)	**Src**	rénlèi gòngyǒu **23** duì rǎnsètǐ
	Trans	There were **23 23** pairs of chromosomes in human beings
(2)	**Src**	qìng xiānggǎng huíguī **5 zhōunián** gōngwùyuán shūhuà dàsài jiāng jǔxíng
	Trans	Chinese civil service calligraphy competition to be held on Hong Kong's return

Table 1. Two examples of Chinese-to-English NMT.

the problem of over-translation and under-translation, we propose a method to involve the bilingual history information into the calculation of attention. The main idea is to gather the translated source and target information for each source word at each time step, and then accumulate the translated bilingual history up to now related to each source word with GRUs. In this way, we can evaluate the completion degree for each source word and give reasonable suggestion for the calculation of attention. Experiments on the Chinese-to-English and English-to-German translation tasks show that our method can achieve significantly improvements over strong baselines and can also produce better alignment.

2 Background

Our work is initially based on the representative attention-based NMT model[1]. The basic framework is a mature end-to-end system following the encoder-decoder framework whose encoder consists of a RNN or bi-directional RNN to generate the representations of the source sentence as a sequence of vectors. The framework employed another RNN network as decoder to learn to align and translate by reading the vectors at the same time. In particular, the framework above possesses an extra attention module which is a mechanism for improving alignment. We'll explain the model and its sub-components in detail in the following section.

Encoder The encoder employs two GRUs to run through the source words bi-directionally and obtain two sequences of hidden states as follows:

$$\overrightarrow{\mathbf{h}}_j = \overrightarrow{\mathbf{GRU}}\left(x_j, \overrightarrow{\mathbf{h}}_{j-1}\right) \tag{1}$$

$$\overleftarrow{\mathbf{h}}_j = \overleftarrow{\mathbf{GRU}}\left(x_j, \overleftarrow{\mathbf{h}}_{j+1}\right) \tag{2}$$

The formal representation of each word in the source sequence is the given by concatenating the corresponding hidden states in both direction, which is shown by Eq.3:

$$\mathbf{h}_j = \left[\overrightarrow{\mathbf{h}}_j; \overleftarrow{\mathbf{h}}_j\right] \tag{3}$$

Attention The design of attention section is inspired by the intuition that corresponding pair of source-end word and target-end word can be highly connected when generating a new word. Thus, the module aims at building direct connections between those highly related source and target words.

Above all, we need to compute the relevance between target word $\mathbf{y}_j$ and $\mathbf{h}_i$, which can be evaluated as

$$e_{ji} = \mathbf{v}_a^T \tanh\left(\mathbf{W}_a \mathbf{s}_{i-1} + \mathbf{U}_a \mathbf{h}_j\right) \tag{4}$$

For computational convenience, we will use following formula to normalize the relevance of $\mathbf{h}_i$ in the source hidden state sequence in j-th decoding step:

$$\alpha_{ji} = \frac{\exp\left(e_{ji}\right)}{\sum_{j'=1}^{l_s} \exp\left(e_{j'i}\right)} \tag{5}$$

Finally, the attention can be compute as weighted summation of all source hidden states by their normalized relevance obtained in the previous step

$$\mathbf{a}_i = \sum\nolimits_{l=1}^{l_s} \alpha_{ji} \mathbf{h}_j \tag{6}$$

where l_s is the length of source inputs. **Decoder** The decoder works by predicting a probability distribution over all the words within the vocabulary and output the target word with the greatest probability. It also use a variant of GRU network to roll the target information, the details of which are described in [1]. Then the current target hidden state s_i is given by

$$\mathbf{s}_i = f(\mathbf{y}_{i-1}, \mathbf{s}_{i-1}, \mathbf{a}_i) \tag{7}$$

The probability distribution $\mathcal{D}_i$ over the target vocabulary at the i-th step depends on the combinational effect of previous ground truth word, the attention $\mathbf{a}_i$ and the rolled target information $\mathbf{s}_i$, the relationship can be described mathematically as

$$\mathbf{t}_i = g(\mathbf{y}_{i-1}, \mathbf{a}_i, \mathbf{s}_i) \tag{8}$$

$$\mathbf{o}_i = \mathbf{W}_o \mathbf{t}_i \tag{9}$$

$$\mathcal{D}_i = \mathrm{softmax}\left(\mathbf{o}_i\right) \tag{10}$$

where g represents a linear transformation, $\mathbf{t}_i$ can be mapped to $\mathbf{o}_i$ by $\mathbf{W}_o$ so that each target word has only one corresponding dimension in $\mathbf{o}_i$.

Intuitively, the probability α_{ji} and the variable e_{ji} jointly reflect the influence of $\mathbf{h}_j$ in deciding next hidden state and even generating next target word.

3 The Proposed Method

The attention component collects source information at each time step by weightedly summing the semantic of all the source words and then the decoder produces a target

word according to the generated attention. In this process, there is a semantic projection between the source attention and the target information. It implies that the semantics held by the source attention and the generated target word is equivalent. Thus we can derive the consumed source semantic and the generated target semantic related to each source word at each step. With this, we can get the accumulated consumed source semantic and generated target semantic up to each time step. The bilingual history semantic can well indicate completion degree of each source word and hence help to generate more reasonable attention.

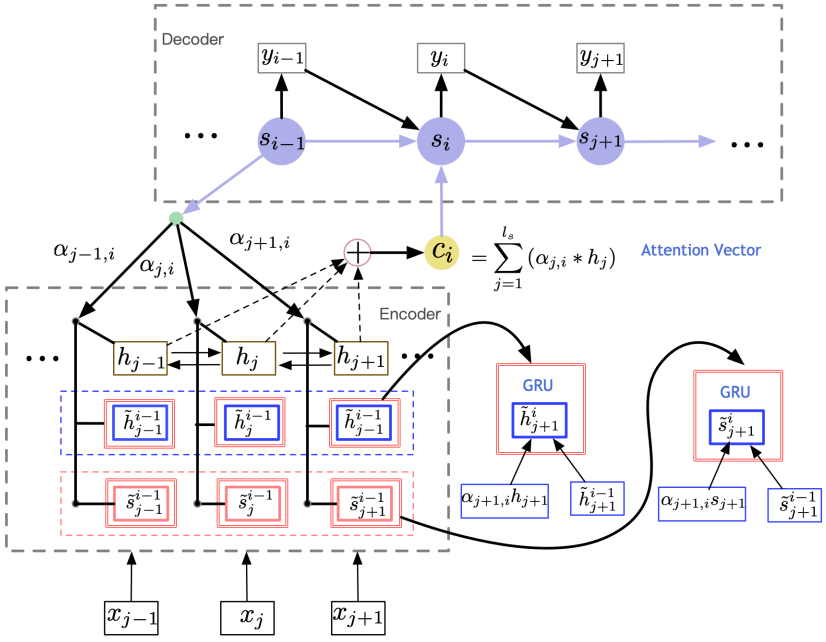

Fig. 1. The architecture of our method with bilingual history involved attention.

Figure 1 gives the architecture of our method. After the target word y is generated y_i, the source information related to the source word x_j is accumulated via a GRU to be $\tilde{\mathbf{h}}_j^i$, and similarly the target information related to the source word x_j is accumulated to $\tilde{\mathbf{s}}_j^i$. Then to generate the next target word y_{i+1}, the accumulated bilingual information is involved to calculated the attention weight of x_j and the weighted sum over the source hidden states is treated as the attention and fed to the decoder.

In this paper, we attempt to add different part of information as

* **SA-NMT**: Only involve the source information up to now in the calculation of attention;
* **TA-NMT**: Only involve the target information up to now in the calculation of attention;

* **BA-NMT**: Involve both the source and target information up to now in the calculation of attention.

3.1 Source History Involved Attention

At the i-th time step, assume the source information related to the source word x_j is $\tilde{\mathbf{h}}_j^{i-1}$. To generate the target word y_i, we calculate the attention with source history information involved and get

$$e_{ji} = \mathbf{v}_a^T \tanh\left(\mathbf{W}_a \mathbf{s}_{i-1} + \mathbf{U}_a \mathbf{h}_j + \mathbf{V}_h \tilde{\mathbf{h}}_j^{i-1}\right) \tag{11}$$

Then we can get the attention following Equation 5 and 6.

According to the attention wight α_{ji} to the source word x_j, we can think at the i-th time step , the quantity of the translated source information related to x_j is

$$\mathbf{I}_{ji}^S = \alpha_{ji} * \mathbf{h}_j \tag{12}$$

But we cannot accumulate the source information related to the source word directly by adding them, as at each time step the translated information is not normalized against the source word. Here we employ a GRU to accumulate it, hoping the learnable update gate and reset gate can perform normalization dynamically. Based on the source information up to the $i-1$-th time step, we can update to get the source information up to the i-th time step related to the word x_j as

$$\tilde{\mathbf{h}}_j^i = \mathbf{GRU}(\mathbf{I}_{ji}^S, \tilde{\mathbf{h}}_j^{i-1}) \tag{13}$$

We initialize $\tilde{\mathbf{h}}_j^0$ with 0, which means that no source words have been translated yet. Besides, the accumulated source information also attention the calculation of logit shown in Equation 8. Before fed to logit, a weighted sum with the attention weights is performed over the history source information related to each source word as

$$\begin{aligned} \tilde{\mathbf{h}}^{i-1} &= \sum_j \alpha_{ji} * \tilde{\mathbf{h}}_j^{i-1} \\ \mathbf{t}_i &= g(\mathbf{y}_{i-1}, \mathbf{a}_i, \mathbf{s}_i, \tilde{\mathbf{h}}^{i-1}) \end{aligned} \tag{14}$$

3.2 Target History Involved Attention

When calculating the attention, it can be considered that the source-side information contained in the current attention is equal to the information of the current generated target word. So each source word corresponds to the current target information:

$$\mathbf{I}_{ji}^T = \alpha_{ji} * \mathbf{s}_{i-1} \tag{15}$$

Then again, $\mathbf{I}_{ji}^T$ is not normalized for the source words, and we still need GRU to accumulate it:

$$\tilde{\mathbf{s}}_j^i = \mathbf{GRU}(\mathbf{I}_{ji}^T, \tilde{\mathbf{s}}_j^{i-1}) \tag{16}$$

where $\tilde{\mathbf{s}}_j^i$ denotes historical information accumulated by the target end. We also take these historical target information into account when calculating attention, so we rewrite the attention model Eq.(4) as follows:

$$e_{ji} = \mathbf{v}_a^T \tanh\left(\mathbf{W}_a \mathbf{s}_{i-1} + \mathbf{U}_a \mathbf{h}_j + \mathbf{V}_s \tilde{\mathbf{s}}_j^{i-1}\right) \tag{17}$$

Note that $\tilde{\mathbf{s}}_j^i$ measures the relevance between the translated historical information of target-end and the corresponding j-th source hidden state. Then, we rewrite the $\mathbf{t}_i$ in Eq.(8) as follows:

$$\begin{aligned} \tilde{\mathbf{s}}^{i-1} &= \sum_j \alpha_{ji} * \tilde{\mathbf{s}}_j^{i-1} \\ \mathbf{t}_i &= g(\mathbf{y}_{i-1}, \mathbf{a}_i, \mathbf{s}_i, \tilde{\mathbf{s}}^{i-1}) \end{aligned} \tag{18}$$

3.3 Bilingual History Involved Attention

Fig.1 illustrates concatenation pattern of the bilingual history involved attention mechanism. The bilingual historical information is the amount of information that has been translated for each source word and the amount of information that has been translated for the target when calculating attention. Intuitively, we combine the bilingual history together by rewriting the attention model. Thus we have

$$\begin{aligned} e_{ji} = \mathbf{v}_a^T \tanh(\mathbf{W}_a \mathbf{s}_{i-1} + \mathbf{U}_a \mathbf{h}_j + \\ \mathbf{V}_h \tilde{\mathbf{h}}_j^{i-1} + \mathbf{V}_s \tilde{\mathbf{s}}_j^{i-1}) \end{aligned} \tag{19}$$

4 Related Work

Attention in neural machine translation[1, 7] is an imperative mechanism to improve the effect of an Encoder + Decoder model based on RNN, which is designed to assign weights to different inputs. Now some new models [13]are proposed to improve the performance of attention mechanism. Some of them[13] integrate the previous attention history into the current attention for better alignment.

Self-attention is another popular mechanism in recent studies. Look-ahead attention proposed by [17] are able to model dependency relationship between distant target words. The model extends the mechanism by referring to previous generated target words, while by and large, previous works focus on learning to align with source words. [5] further presented a variational self-attention mechanism extracts different aspects of the sentence and partition them into multiple vector representations.

Exploiting historical information to improve the performance of Attention is also a novel mechanism. [8] proposed to introduce source-end historical information onto attention, which use interactive attention to rewrite the source information during translation. Interactive attention to keep tracking the source history by reading and writing operations. [16] proposed to introduce target-end historical information onto attention, which focuses on integrating the decoding history. However, the utilization of historical information basically limited to either source-end or target-end by then, our work managed to combine bilingual history together.

5 Experiments

5.1 Data Preparation

We mainly evaluated our approach on the widely used NIST Chinese-English translation task. In addition, to show the usefulness of our approach, we also provided the results of the English-German translation task. So we carried out experiments on two datasets:

NIST Zh→En: Our training data for the Chinese-English training task consists of

Systems	**MT03**	**MT04**	**MT05**	**MT06**	**Average**
RNNsearch	35.75	38.68	34.69	37.61	*36.68*
RNNsearch*	42.03	44.58	42.33	42.40	*42.84*
NN-Coverage	42.69	44.92	42.74	42.79	*43.29*
IA-Model	42.83	45.14	42.94	43.12	*43.51*
Transformer-base	44.56	45.81	44.12	43.31	*44.45*
BA-NMT	**43.73**‡	**45.77**‡*	**43.58**‡*	**43.91**‡*	***44.25 +1.41***

Table 2. Performance comparison on Zh→En translation. The "‡" indicates statistically significant improvement over RNNsearch*. "$\star$" means statistically significant improvement over NN-Coverage and IA-Model. Here $\rho < 0.05$ [14].

1.25M sentence pairs[3]. We chose the NIST 2002 test set as our development set, and the NIST 2003, 2004, 2005, 2006 datasets as the test sets.

WMT14 En→De: Our training data for the English-German training task consists of 4.45M sentence pairs. We use newstest2013 as the valid set, and newstest2014 as the test set.

In our experiments, we used the case-insensitive 4-gram BLEU[10] for **Zh→En** and case-sensitive for **En→De** to evaluate the translation performance.

5.2 Systems

We involved following systems as below:

RNNsearch We implemented the conventional attention-based Neural Machine Translation of [1] with PyTorch [4].

RNNsearch* This is an improved system of RNNsearch, the detail we can see in this link[5].

NN-Coverage A variants of attention-based NMT model [13] which maintain a soft coverage on each source representation to keep track of the history to improve the attention mechanism.

IA-Model An improved NMT model which can capture translation status with an interactive attention to track attention history.

[3] These sentence pairs are mainly extracted from LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06

[4] `http://pytorch.org`

[5] `https://github.com/nyu-dl/dl4mt-tutorial`

5.3 Configuration

For the NIST Zh→En data set, we adopted 16k byte pair encoding (BPE) merging operations [11] in the source and target end, respectively. The length of the sentences was limited up to 128 tokens on both ends. For WMT En→De, the number of merge operations in BPE is set to 32K for both source and target languages, and the maximum length of sentences in the En→De task is also set to 128.

We deployed shared configuration for all the systems. All the embedding sizes were both set to 512, the size of all hidden units in encoder and decoder RNNs was also set to 512, and all parameters were initialized by using uniform distribution over $[-0.1, 0.1]$. The mini-batch stochastic gradient descent (SGD) algorithm was employed. We batch sentence pairs according to the approximate length, and limit input and output tokens to 4096. In addition, the learning rate was adjusted by adam optimizer [4]($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e^{-6}$). Dropout was applied on the output layer with dropout rate of 0.2. The beam size was set to 10.

5.4 Ablation Study

Systems	**Zh→En**
RNNsearch	36.68
RNNsearch*	42.84
+ SA-NMT	43.52
+ TA-NMT	43.83
+ BA-NMT	**44.25**

Table 3. Ablation study with average BLEU scores.

Systems	**En→De**
RNNsearch*	25.76
+ SA-NMT	26.11
+ TA-NMT	26.32
+ BA-NMT	**26.58**

Table 4. Performance comparison on En→De translation.

We employed several methods to improve the performance of our model. For instance, we keep track of source history and put it into attention model, which settles the problem of missing translation to a certain extent. Furthermore, we model the dependency relationship between the previous generated target words and the source words where each pair of source word and generated target word is one-to-one correspondence.

The translation performance is listed in Table 3 measuring in BLEU score. It is obvious that in all the cases, our proposed history involved attention model outperforms RNNsearch* system. Specifically, we obtained a BLEU score of 43.52 when only employing the Source History Involved Attention, which indicated that feeding predicted

words as context can sufficiently mitigate exposure bias. In comparison, we improved RNNsearch* by 0.68 BLEU points, which also proves its effectiveness. Likewise, we are also gratified by the result of only applying Target History Involved Attention, which achieved a comparable BLEU score as Source History Involved Attention, we improved RNNsearch* by 0.99 BLEU points. Eventually, we managed to combine the above two attention mechanism together and expect to get a more remarkable improvement.

On the En-De dataset, as shown in Table 4, BA-NMT shows superiority on test dataset, and achieves the gains of 0.8 BLEU points over RNNsearch* system. Given the above results, we can conclude that BA-NMT can indeed better utilize the historical information and bring improvement on the translation performance.

5.5 Alignment Quality

As the results of BLEU scores have proved that our method can achieve more accurate translation, we then try to verify this conclusion from another perspective. Since there is a common belief that the better translation should have better alignment with the source sentence, intuitively, we try to evaluated the quality of the alignments derived from the attention module of NMT using AER [9]. As for dataset, we consider the human aligned dataset from [6], containing 900 Chinese-English sentence pairs, to evaluate alignment quality in our experiment.

In practice, we adopted the method that retain the alignment link with the highest probability in Eq.(5). As a comparison, we report the results of both the baseline system and our system. Measured by BLEU score, the results shown in Table 5 illustrate that our system BA-NMT is able to produce more accurate translation than the RNNsearch*. Meanwhile, our corresponding AER score is lower, suggesting better alignments.

SYSTEMS	**BLEU**	**AER**
RNNsearch*	42.84	44.03
BA-NMT	**44.25**	**42.16**

Table 5. Comparison of alignment quality on Zh→En translation task, the BLEU and AER scores are evaluated on different test sets.

6 Conclusion

In this work, we demonstrate a novel Bilingual History Involved Attention for the attention-based NMT. Our core innovation is that our model allows to maintain track of both the target history and the source history, which is beneficial for our model to better utilize the historical information and generate more accurate translation. We further explore the application of our model on NMT tasks and conduct experiments by using three strategies to integrate the historical information into NMT. Results of empirical studies are consistent with our expectation, which proves that our Bilingual History Involved Attention model is capable of achieving better alignment quality than baseline model, especially in the complicated cases. Besides, the proposed model could effectively alleviated the problem of over-translation and under-translation.

References

1. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR 2015*.
2. 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*.
3. Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.
4. Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
5. Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.
6. Yang Liu and Maosong Sun. 2015. Contrastive unsupervised word alignment with non-local features. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2295–2301. AAAI Press.
7. Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
8. Fandong Meng, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Interactive attention for neural machine translation. *arXiv preprint arXiv:1610.05011*.
9. Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.
10. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.
11. Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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. Association for Computational Linguistics.
12. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. 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.
13. Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811*.
14. Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 531–540. Association for Computational Linguistics.
15. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.
16. Mingxuan Wang, Jun Xie, Zhixing Tan, Jinsong Su, Deyi Xiong, and Chao Bian. 2018. Neural machine translation with decoding history enhanced attention. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1464–1473.
17. Long Zhou, Jiajun Zhang, and Chengqing Zong. 2017. Look-ahead attention for generation in neural machine translation. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 211–223. Springer.