Cross Aggregation of Multi-Head Attention for Neural Machine Translation

Juncheng Cao, Hai Zhao*, and Kai Yu

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China
caojuncheng@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn, kai.yu@sjtu.edu.cn

Abstract. Transformer based encoder has been the state-of-the-art model for the latest neural machine translation, which relies on the key design called self-attention. Multi-head attention of self-attention network (SAN) plays a significant role in extracting information of the given input from different subspaces among each pair of tokens. However, that information captured by each token on a specific head, which is explicitly represented by the attention weights, is independent from other heads and tokens, which means it does not take the global structure into account. Besides, since SAN does not apply an RNN-like network structure, its ability of modeling relative position and sequential information is weakened. In this paper, we propose a method named Cross Aggregation with an iterative routing-by-agreement algorithm to alleviate these problems. Experimental results on the machine translation task show that our method help the model outperform the strong Transformer baseline significantly.

Keywords: Machine translation · Attention mechanism · Information aggregation.

1 Introduction

Traditional attention mechanism was first introduced in the field of neural machine translation by Bahdanau et al. [1] and then its variants quickly become the essential technique in achieving promising performances in various of tasks such as document classification [45], speech recognition [6] and many other applications. Although the neural machine translation has witnessed a revolutionary performance improvement with the use of attention mechanism, most work focused on a recurrent neural network (RNN) structure e.g. LSTM [12] or GRU [5] which cannot support parallel computation conveniently.

In order to address the problem, Vaswani et al. [32] proposed a multi-head attention mechanism in SAN, which can on one hand support efficiently parallel computation and on the other hand further improve the performance of neural machine translation. The

* Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100) and key projects of National Natural Science Foundation of China (No. U1836222 and No. 61733011).

basic idea of multi-head attention is to parallelly capture linguistic information which have been transformed into multiple distinct subspaces with simple linear transformation functions.

Most existing work based on multi-head attention tend to obtain a better partial representation on different heads [23], some other studies focus on the information aggregation across the SAN, e.g. Dou et al. [7] aggregate the hidden states output in different layers of Transformer encoder as partial input of the decoder. While the existing methods of information aggregation of SAN do not pay much attention to the lack of positional information, and that is an obvious limitation of SAN's performance, which it has to implement a positional embedding method to alleviate. Besides, since the input sequence is transformed into multi-dimensional space, aggregating method should naturally conducted from different directions, which is not seen in the previous work.

In this paper, we propose a method named *Cross Aggregation* to aggregate global context information in two directions cross with each other. We choose to leverage the basic algorithm framework of routing-by-agreement [28] with some multi-head-attention-based features to solve the problems mentioned above. Basically, the algorithm is to address the problem of assigning different parts different weights to construct a final whole output. It is implemented in an iterative way to dynamically update all the weights with quite simple parallel computations which can benefit from GPU acceleration.

We evaluate the performance of our proposed aggregating method on two widely-used translation tasks: WMT17 Chinese-to-English and WMT14 English-to-German. Experimental results demonstrate that our method have better performance over the strong Transformer baseline [32] and other existing NMT models.

2 Background

Attention mechanism was designed to model the different weights between an output representation and multiple input representations, which reflects the relevance between the output and each part of input. Recently, Vaswani et al. [32] proposed a multi-head attention mechanism, which benefits from capturing context relevance information in multiple subspaces with different heads, where each head represents an individual transformation function.

Formally, given the input of query $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_L]$, key-value pairs $\{\mathbf{K}, \mathbf{V}\} = \{(\mathbf{k}_1, \mathbf{v}_1), \ldots, (\mathbf{k}_M, \mathbf{v}_M)\}$, where $\mathbf{Q} \in \mathbb{R}^{L\times d}$, $\{\mathbf{K}, \mathbf{V}\} \in \mathbb{R}^{M\times d}$. d denotes the dimensionality of the hidden states. The output is mapped from $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$. In multi-head attention, if there are H heads, the $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ will be transformed into H subspaces by individual learnable linear transformation matrix:

$$\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h = \mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}\mathbf{W}_h^K, \mathbf{V}\mathbf{W}_h^V \tag{1}$$

where $\mathbf{Q}_h$, $\mathbf{K}_h$, and $\mathbf{V}_h$ are the transformed representations of h-th head of query, key and value. The transformation matrices $\{\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V\} \in \mathbb{R}^{d\times\frac{d}{H}}$. On each head, it will apply a attention function $\textsc{Att}(\cdot)$ over the query and the key, then calculate the weighted average on the value to obtain the partial output:

$$\mathbf{O}_h = \textsc{Att}(\mathbf{Q}_h, \mathbf{K}_h)\mathbf{V}_h \tag{2}$$

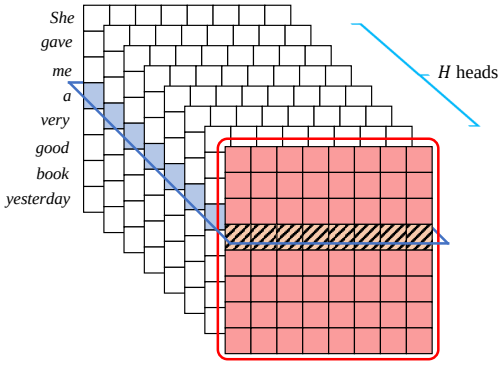

Fig. 1. Vertical and horizontal capsules. This illustration shows the matrix form of the attention results on H heads. The red block represents the vertical capsule $\mathbf{E}_h^{\updownarrow}$, and the blue block represents the horizontal capsule $\mathbf{E}_l^{\leftrightarrow}$. The shadowed orange part is their overlapping attention vector $\mathbf{e}_{l,h}$.

where $\mathbf{O}_h \in \mathbb{R}^{L \times \frac{d}{H}}$. In this paper, we apply the scaled dot-product attention [24] which achieves promising performance and suitable for parallel computing in practice [32]:

$$\textsc{Att}(\mathbf{Q}_h, \mathbf{K}_h) = softmax(\mathbf{E}_h) \tag{3}$$

$$\mathbf{E}_h = \frac{\mathbf{Q}_h \mathbf{K}_h^T}{\sqrt{d}} = [\mathbf{e}_{1,h}, \ldots, \mathbf{e}_{L,h}] \tag{4}$$

$$\mathbf{e}_{l,h} = \frac{\mathbf{q}_l \mathbf{W}_h^Q \mathbf{K}_h^T}{\sqrt{d}} \in \mathbb{R}^M, l = 1, \ldots L \tag{5}$$

where $\mathbf{e}_{l,h}$ is the attention vector of the l-th query token on the h-th head.

3 Approach

3.1 Information Aggregation with Capsule Routing

The goal of our work is to aggregate information of other heads and tokens onto each specific attention vector so that the attention weights can be further adjusted according to the global structure and sequential information. Therefore, an iterative algorithm called *routing-by-agreement* applied in the capsule network [28] is suitable for the goal. Concretely speaking, The basic idea of that algorithm is to iteratively decide the weight of each part which will be gathered as the final whole output.

In capsule network, one capsule means a group of neurons, and different capsules can be viewed as the representations of one single entity individually from multiple perspectives or directions. It was first proposed and applied in the field of computer vision and it is intuitively for us to find that the multi-head attention mechanism has a similar structure. We can therefore view any specific attention vector $\mathbf{e}_{l,h}$ as a part of two separate capsules: 1) capsule that consists of all the attention vectors on the h-th head, 2) capsule that consists of attention vectors of l-th token on all the H heads. Thus, as

Algorithm 1 Iterative Simple Routing

Require: $L \times N$ vote vectors $\mathbf{V}_{l \to n}$, iteration times T
Ensure: N output capsules $\mathbf{\Omega}_n$
1: **function** SIMPLEROUTING($\mathbf{V}, T$)
2: $\forall \mathbf{V}_{l \to n}$: initialize $B_{l \to n}$
3: **for** T iterations **do**
4: $\forall (l \to n)$: $C_{l \to n} \leftarrow softmax(B_{* \to n})$
5: $\forall \mathbf{\Omega}_n$: compute $\mathbf{\Omega}_n$ by Eq. 7
6: $\forall (l \to n)$: $B_{l \to n} \mathrel{+}= \mathbf{\Omega}_n \cdot \mathbf{V}_{l \to n}$
7: **end for**
8: **return** $\mathbf{\Omega}$
9: **end function**

shown in Figure 1, in a matrix way, we call these two types of capsules *vertical capsules* $\mathbf{E}_h^{\updownarrow} \in \mathbb{R}^{L \times M}$ and *horizontal capsules* $\mathbf{E}_l^{\leftrightarrow} \in \mathbb{R}^{H \times M}$ according to their arrangement directions, repectively.

3.2 Routing-by-Agreement

In this work, we apply the routing-by-agreement algorithm proposed in paper [28] named *simple routing* for the information aggregation task.

Formally, the routing algorithm has two layers which called *input capsules* and *output capsules*. Given N output capsules, each input capsule should have exactly N corresponding *vote vectors* to measure the relevance between input capsule and the associated output capsule. More specifically speaking, given L input capsules $\{\mathbf{H}_1, \ldots, \mathbf{H}_L\}$, we have $L \times N$ vote vectors calculated by:

$$\mathbf{V}_{l \to n} = \mathbf{H}_l \mathbf{W}_{l \to n} \tag{6}$$

For each vote vector $\mathbf{V}_{l \to n}$, we maintain a dynamically updated weight $C_{l \to n}$. The final output capsule $\mathbf{\Omega}_n$ is calculated by:

$$\mathbf{\Omega}_n = \frac{\|\mathbf{S}_n\|^2}{1 + \|\mathbf{S}_n\|^2} \frac{\mathbf{S}_n}{\|\mathbf{S}_n\|} \tag{7}$$

$$\mathbf{S}_n = \sum_{l=1}^{L} C_{l \to n} \mathbf{V}_{l \to n} \tag{8}$$

where Eq. 7 is a non-linear function called "squashing" function in paper [28].

Algorithm 1 shows the detail of iterative simple routing mechanism. $B_{l \to n}$ are set to measure the degree in which one input capsule participates in the constructing of the final output capsule, and they are initialized as all zero (line 2). To update the dynamic weight $C_{l \to n}$, it computes the softmax of all the $B_{l \to n}$ associated with $\mathbf{\Omega}_n$ in the current iteration.

3.3 Cross Aggregation

As shown in Figure 1, each specific attention vector $\mathbf{e}_{l,h}$ belongs to two groups of neurons, i.e., capsules which are cross with each other. And *cross aggregation* aims to aggregate information in these two dimensions onto their overlapping attention vector with simple routing algorithm. Formally, we add the vertical and horizontal output capsules to the original attention matrix $\mathbf{E}$, i.e., $\widehat{\mathbf{E}} = \mathbf{E} + \mathbf{\Omega}^{\updownarrow} + \mathbf{\Omega}^{\leftrightarrow}$. so that the Eq. 3 is rewritten as:

$$\widehat{\textsc{Att}}(\mathbf{Q}, \mathbf{K}) = softmax(\widehat{\mathbf{E}}) \tag{9}$$

And we argue that in the scenario of multi-head attention, each $\mathbf{e}_{l,h}$ itself can naturally be the so-called vote vector so that we do not apply a learnable linear transformation matrices as in the vanilla algorithm.

Vertical Capsule $\mathbf{E}_h^{\updownarrow}$ Since one vertical capsule has L vote vectors when the input query has that length, we will therefore obtain L output vertical capsules through the simple routing algorithm:

$$\mathbf{V}_{h\to l}^{\updownarrow} = \mathbf{e}_{l,h} \tag{10}$$

$$\widetilde{\mathbf{\Omega}}^{\updownarrow} = \textsc{SimpleRouting}(\{\mathbf{E}_h^{\updownarrow}\}, T) \in \mathbb{R}^{L\times M} \tag{11}$$

In previous work [26, 27], the multi-layer SAN was found having a hierarchical feature that it captures lexical information in the lower layers while higher layers tend to learn semantical information. Therefore we consider that the same head in different layers will accept the global information in different degrees. To measure the acceptance extent we simply assign a learnable weight for each head in each layer based on their voting weights on the final iteration stage:

$$\mathbf{\Omega}^{\updownarrow} = [\lambda_1^{\updownarrow}\widetilde{\mathbf{\Omega}}^{\updownarrow}, \ldots, \lambda_H^{\updownarrow}\widetilde{\mathbf{\Omega}}^{\updownarrow}] \tag{12}$$

$$\mathbf{\Lambda}^{\updownarrow} = softmax(\mathbf{W}^{\updownarrow}[\sum_{l=1}^{L} B_{1\to l}, \ldots, \sum_{l=1}^{L} B_{H\to l}]) \tag{13}$$

Horizontal Capsule $\mathbf{E}_l^{\leftrightarrow}$ Basically, the processing method of L horizontal capsules $\{\mathbf{E}_l^{\leftrightarrow}\}$ can be similar with that of the vertical capsules, i.e., assign $\mathbf{V}_{l\to h}^{\leftrightarrow} = \mathbf{e}_{l,h}$ for each horizontal capsule $\mathbf{E}_l^{\leftrightarrow}$ and apply the simple routing algorithm.

However, in this way it will omit some essential features that are not owned by vertical capsules. Therefore we here propose two methods: positional capsule routing and self initialization.

Positional Capsule Routing Different from vertical capsules which are order independent, the position arrangement of L horizontal capsules contains the sequential information of the input hidden states. Therefore, simply aggregating all the horizontal capsules without considering the inner order of the sequence will only make it become a complicated bag-of-words model. To let the model be aware of that inner order, we propose the positional capsule routing method.

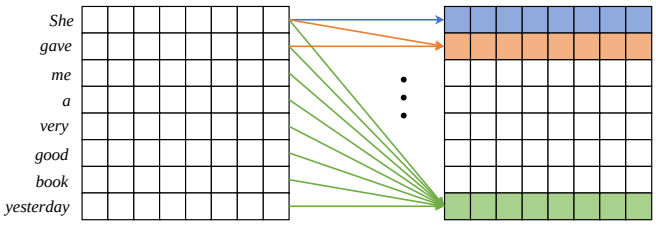

Fig. 2. Positional Capsule Routing

As shown in Figure 2, for each token of the input query hidden states we apply a partial simple routing algorithm to obtain the corresponding $\widetilde{\mathbf{\Omega}}^{\leftrightarrow}$. Concretely speaking, for each specific horizontal capsule, we only apply the aggregation on the capsule set that excludes the capsules below itself, which means the tokens appear relative later in the input sequence will not be aggregated:

$$\begin{aligned} \widetilde{\mathbf{\Omega}}_l^{\leftrightarrow} &= \textsc{SimpleRouting}(\{\mathbf{E}_{t\leq l}^{\leftrightarrow}\}, T) \in \mathbb{R}^{H\times M} \\ \mathbf{\Omega}^{\leftrightarrow} &= [\widetilde{\mathbf{\Omega}}_1^{\leftrightarrow}, \ldots, \widetilde{\mathbf{\Omega}}_L^{\leftrightarrow}] \end{aligned} \tag{14}$$

Here the reason we do not further apply a similar "backward" positional routing on those horizontal capsules is that if we calculate both the forward and backward output capsules, it would be confused for the network to determine the real token order. Since each final output capsule $\widetilde{\mathbf{\Omega}}_l^{\leftrightarrow}$ would therefore come from two sources, forward and backward, and for the corresponding l-th token, it would be hard to tell which part some other token belongs to.

Self Initialization In the vanilla version of simple routing, the weights of vote vectors are all assigned zero at the initialization phase of the algorithm. One explanation for doing so is that for a general aggregation task, we do not have prior knowledge about the possible weight distribution of the aggregated parts, otherwise we could initialize them with different values. Here in the situation of SAN, we expect it would be naturally that each element of the attention vector $\mathbf{e}_{l,h}$ measures the prior voting weight for each token pair. More specifically, in the multi-head attention network of encoder, where the $\mathbf{Q} = \mathbf{K} = \mathbf{V}$, the attention weight of the l-th token to the m-th token on the h-th head $\alpha_{l,m}^h$ itself, which we think, can be the initialization weight when computing the output capsule for the l-th token. Therefore the initialized weight is calculated by:

$$\begin{aligned} B_{t\to h}^{init} &= \alpha_{l,t}^h, t \leq l \\ \alpha_{l,t}^h &= \frac{\mathbf{q}_l \mathbf{W}_h^Q (\mathbf{k}_t \mathbf{W}_h^K)^T}{\sqrt{d}} \end{aligned} \tag{15}$$

when applying Eq. 14.

#	Model	BLEU	Δ
1	Transformer-Base	24.28	-
2	+ Horizontal w/ Zero Initialization	24.76	+0.48
3	+ Horizontal w/ Self Initialization	$24.88^{\uparrow}$	+0.60
4	+ Vertical + Horizontal w/ Zero Initialization	$25.02^{\Uparrow}$	+0.74
5	+ Vertical + Horizontal w/ Self Initialization	24.68	+0.40

Table 1. Translation performances of model variations on WMT17 Chinese-to-English (Zh⇒En) task. "↑ / ⇑": significantly better than the baseline counterpart ($p < 0.05/0.01$).

4 Experiment

4.1 Setup

We conduct experiments on widely-used WMT17 Chinese-to-English (Zh⇒En) and WMT14 English-to-German (En⇒De) datasets. For Zh⇒En task, the parallel corpus dataset contains total 20.6M sentence pairs, but we only keep those with the sentence length less than 50. The newsdev2017 is used as the validation set and the newstest2017 as the test set through the training process. While for En⇒De task, the dataset consists of 4.6M sentence pairs, and we choose newstest2013 as the validation set and newstest2014 is used to test the model performance. We employ byte-pair encoding (BPE) [29] and set the merge operations as 32K for both WMT17 and WMT14 in order to reduce the vocabulary size.

We implement our proposed approach on the Transformer model [32]. The model Transformer-*Base* and *Big* differ at word embedding size (512 vs 1024), feed-forward network dimensionality (2048 vs 4096) and the number of attention heads (8 vs 16). The dropout rate is changed from 0.1 to 0.3 when training the *Big* model compared to the *Base* one. We follow their parameter configuration of the *Base* model to train our baseline on both Zh⇒En and En⇒De tasks. We set batch size to 2048 tokens and the gradient accumulation times to 12 before the back-propagation. We use the OpenNMT-py framework [14] to implement our method and use the case-sensitive 4-gram NIST BLEU score [25] as the metric to evaluate our models. All the model trainings are on two NVIDIA GeForce GTX 1080 Ti GPUs.

Empirically, we set the parameter iteration times T of all the models using the aggregation method with the number 3. In previous work [7, 16], researchers find that the overall performance of the model can achieve the best when iteration times T is set to 3. This result is also consistent with the findings in paper [28]. In this work, we find that over half of the vote vectors' weights come out to be zero which causes a worse performance when we set the iteration times to 4 or 5.

4.2 Results

Model Variations Table 1 shows the translation results on the WMT17 Chinese-to-English task. From the table we can see that all the models that apply the aggregation methods we propose in this paper consistently outperform the baseline model, which

System	Architecture	Zh⇒En	En⇒De
Existing NMT Systems			
Wu et al. [40]	RNN with 8 layers	N/A	26.30
Gehring et al. [8]	CNN with 15 layers	N/A	26.36
Vaswani et al. [32]	Transformer-Base	N/A	27.30
	Transformer-Big	N/A	28.40
Hassan et al. [10]	Transformer-Big	24.20	N/A
Li et al. [16]	Transformer-Base + Effective Aggregation	24.68	27.98
Our NMT Systems			
this work	Transformer-Base	24.28	27.43
	+ Cross Aggregation	**25.02**	**28.04**

Table 2. Comparing with existing NMT systems on WMT17 Chinese-to-English (Zh⇒En) and WMT14 English-to-German (En⇒De) tasks.

demonstrates the effectiveness of the cross aggregation mechanism. Frow row 2 we can see that simply applying the positional horizontal routing will improve the performance up to +0.48 BLEU points, showing that the SAN benefits from capturing more sequential information. Comparing with the row 2 and 3, the +0.12 BLEU points improvement indicates that our approach of self initialization does help the horizontal aggregation to calculate assigned weights more reliably.

The cross aggregation with zero initialization (Row 4) achieves the highest score with a +0.74 BLEU points improvement while the self initialization counterpart (Row 5) the lowest. On one hand it does demonstrate the superiority of our cross aggregation mechanism, on the other hand it also indicates that the self initialization method and the vertical routing will influence each other in bad way.

We here try to give an explanation about why the self initialization and the vertical routing fail to be complementary to each other (Row 5). Before we introduce the vertical routing into the attention process, the weights which are used to initialize the horizontal routing on higher layers partially might model the context information among the heads on the lower layers, which means it could roughly play the role of vertical routing and help improve the model performance (Row 3). While with the introduction of vertical routing, the simplicity of self initialization might on the contrary affect the model's capability of capturing context information.

Main Results Table 2 lists the overall result on both WMT17 Chinese-to-English (Zh⇒En) and WMT14 English-to-German (En⇒De) tasks. As shown in the table, cross aggregation approach consistently improves the performance on this two language pairs. For WMT17 Chinese-to-English task, our approach outperforms all the other models above, and for WMT14 English-to-German task, we only inferior to the vanilla Transformer-Big model whose number of parameters is three times more than ours. This shows the effectiveness of our proposed method.

5 Related Work

With the development of research of neural network recently, this advanced method has been applied to several tasks in the field of natural language processing with impressive results e.g. semantic role labeling [4, 20, 11, 9, 22], sentence parsing [15, 39, 18, 52, 19, 21], word segmentation [2, 3, 37], reading comprehension [48, 51, 50], relation extraction [17], IME [13, 49], and researchers also reaches huge success when it comes to NMT [41, 47, 38, 33, 35, 36, 34, 46].

Basically our work is related to the attention optimization of SAN in Transformer. More specifically, the NMT model leverages some extra information to help reach out a better attention value distribution. To alleviate the weakness of Transformer caused by the lack of positional information, it is natural to make the model be aware of the relative position of source input [31, 30, 43]. According to [42], context information through all the layers can help improve the performance of SAN. Combining the layer- and sentence-level information to sharpen the attention result has been proved effective in the final performances [44]. All these work above show that optimizing the attention result with extra information is promising in further research.

6 Conclusion

Inspired by the idea of routing algorithm in capsule network, in this paper we propose a cross aggregation method aiming to capture the global context in two dimensions for the attention score to enhance the state-of-the-art neural machine translation. Our study shows aggregating information from all the heads and tokens is an effective way to improve the attention results and beside the conventional head-wise pattern, provide a novel way to understand the multi-head attention network. Our work also proves that adding positional information into the self-attention network can efficiently strengthen the model ability of capturing relative sequential relationship. Experimental results on two widely-used datasets demonstrate the superiority of our proposed approach.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015) (2015)
2. Cai, D., Zhao, H.: Neural word segmentation learning for Chinese. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016). pp. 409–420 (2016)
3. Cai, D., Zhao, H., Zhang, Z., Xin, Y., Wu, Y., Huang, F.: Fast and accurate neural word segmentation for Chinese. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017). pp. 608–615 (2017)
4. Cai, J., He, S., Li, Z., Zhao, H.: A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018). pp. 2753–2765 (2018)

5. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014). pp. 1724–1734 (2014)
6. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2015). pp. 577–585 (2015)
7. Dou, Z.Y., Tu, Z., Wang, X., Wang, L., Shi, S., Zhang, T.: Dynamic layer aggregation for neural machine translation with routing-by-agreement. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019). pp. 86–93 (2019)
8. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: Proceedings of the 34th International Conference on Machine Learning (ICML 2017). pp. 1243–1252 (2017)
9. Guan, C., Cheng, Y., Zhao, H.: Semantic role labeling with associated memory network. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019). pp. 3361–3371 (2019)
10. Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M., et al.: Achieving human parity on automatic chinese to english news translation. arXiv preprint arXiv:1803.05567 (2018)
11. He, S., Li, Z., Zhao, H., Bai, H.: Syntax for semantic role labeling, to be, or not to be. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018). pp. 2061–2071 (2018)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
13. Huang, Y., Zhao, H.: Chinese pinyin aided IME, input what you have not keystroked yet. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). pp. 2923–2929 (2018)
14. Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.M.: OpenNMT: Open-source toolkit for neural machine translation. arXiv preprint arXiv:1701.02810 (2017)
15. Li, H., Zhang, Z., Ju, Y., Zhao, H.: Neural character-level dependency parsing for chinese. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018). pp. 5205–5212 (2018)
16. Li, J., Yang, B., Dou, Z.Y., Wang, X., Lyu, M.R., Tu, Z.: Information aggregation for multi-head attention with routing-by-agreement. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019). pp. 3566–3575 (2019)
17. Li, P., Zhang, X., Jia, W., Zhao, H.: GAN driven semi-distant supervision for relation extraction. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019). pp. 3026–3035 (2019)
18. Li, Z., Cai, J., He, S., Zhao, H.: Seq2seq dependency parsing. In: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018). pp. 3203–3214 (2018)
19. Li, Z., Cai, J., Zhao, H.: Effective representation for easy-first dependency parsing. In: PRICAI 2019: Trends in Artificial Intelligence - 15th Pacific Rim International Conference on Artificial Intelligence (2019)
20. Li, Z., He, S., Cai, J., Zhang, Z., Zhao, H., Liu, G., Li, L., Si, L.: A unified syntax-aware framework for semantic role labeling. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). pp. 2401–2411 (2018)

21. Li, Z., He, S., Zhang, Z., Zhao, H.: Joint learning of POS and dependencies for multilingual universal dependency parsing. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies (CoNLL 2018). pp. 65–73 (2018)
22. Li, Z., He, S., Zhao, H., Zhang, Y., Zhang, Z., Zhou, X., Zhou, X.: Dependency or span, end-to-end uniform semantic role labeling. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019). pp. 6730–6737 (2019)
23. Lin, Z., Feng, M., Santos, C.N.d., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. In: Proceedings of the 5th International Conference on Learning Representations (ICLR 2017) (2017)
24. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015). pp. 1412–1421 (2015)
25. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002). pp. 311–318 (2002)
26. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018). pp. 2227–2237 (2018)
27. Raganato, A., Tiedemann, J.: An analysis of encoder representations in transformer-based machine translation. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 287–297 (2018)
28. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Proceedings of the 31th Conference on Neural Information Processing Systems (NIPS 2017). pp. 3856–3866 (2017)
29. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016). pp. 1715–1725 (2016)
30. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018). pp. 464–468 (2018)
31. Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., Zhang, C.: DiSAN: Directional self-attention network for rnn/cnn-free language understanding. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018). pp. 5446–5455 (2018)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31th Conference on Neural Information Processing Systems (NIPS 2017). pp. 5998–6008 (2017)
33. Wang, R., Finch, A., Utiyama, M., Sumita, E.: Sentence embedding for neural machine translation domain adaptation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017). pp. 560–566 (2017)
34. Wang, R., Utiyama, M., Finch, A., Liu, L., Chen, K., Sumita, E.: Sentence selection and weighting for neural machine translation domain adaptation. IEEE/ACM Transactions on Audio, Speech, and Language Processing **26**(10), 1727–1741 (2018)
35. Wang, R., Utiyama, M., Liu, L., Chen, K., Sumita, E.: Instance weighting for neural machine translation domain adaptation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). pp. 1482–1488 (2017)
36. Wang, R., Utiyama, M., Sumita, E.: Dynamic sentence sampling for efficient training of neural machine translation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018). pp. 298–304 (2018)

37. Wang, X., Cai, D., Li, L., Xu, G., Zhao, H., Si, L.: Unsupervised learning helps supervised neural word segmentation pp. 7200–7207 (2019)
38. Wu, Y., Zhao, H.: Finding better subword segmentation for neural machine translation. In: The 17th China National Conference on Computational Linguistics (CCL 2018). pp. 53–64 (2018)
39. Wu, Y., Zhao, H., Tong, J.J.: Multilingual universal dependency parsing from raw text with low-resource language enhancement. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies (CoNLL 2018). pp. 74–80 (2018)
40. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
41. Xiao, F., Li, J., Zhao, H., Wang, R., Chen, K.: Lattice-based transformer encoder for neural machine translation. In: Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019). pp. 3090–3097 (2019)
42. Yang, B., Li, J., Wong, D.F., Chao, L.S., Wang, X., Tu, Z.: Context-aware self-attention networks. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019). pp. 387–394 (2019)
43. Yang, B., Tu, Z., Wong, D.F., Meng, F., Chao, L.S., Zhang, T.: Modeling localness for self-attention networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). pp. 4449–4458 (2018)
44. Yang, B., Wang, L., Wong, D.F., Chao, L.S., Tu, Z.: Convolutional self-attention networks. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019). pp. 4040–4045 (2019)
45. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016). pp. 1480–1489 (2016)
46. Zhang, H., Zhao, H.: Minimum divergence vs. maximum margin: an empirical comparison on seq2seq models. In: Proceedings of the 7th International Conference on Learning Representations (ICLR 2019) (2019)
47. Zhang, Z., Wang, R., Utiyama, M., Sumita, E., Zhao, H.: Exploring recombination for efficient decoding of neural machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). pp. 4785–4790 (2018)
48. Zhang, Z., Huang, Y., Zhao, H.: Subword-augmented embedding for cloze reading comprehension. In: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018). pp. 1802–1814 (2018)
49. Zhang, Z., Huang, Y., Zhao, H.: Open vocabulary learning for neural Chinese pinyin IME. In: Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019). pp. 1584–1594 (2019)
50. Zhang, Z., Huang, Y., Zhu, P., Zhao, H.: Effective character-augmented word embedding for machine reading comprehension. In: 7th CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC 2018). pp. 27–39 (2018)
51. Zhang, Z., Zhao, H., Ling, K., Li, J., Li, Z., He, S.: Effective subword segmentation for text comprehension. IEEE/ACM Transactions on Audio, Speech, and Language Processing (2019)
52. Zhou, J., Zhao, H.: Head-driven phrase structure grammar parsing on Penn treebank. In: Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019). pp. 2396–2408 (2019)