Attention over Heads: A Multi-Hop Attention for Neural Machine Translation

Shohei Iida[†], **Ryuichiro Kimura**[†], **Hongyi Cui**[†], **Po-Hsuan Hung**[†], **Takehito Utsuro**[†] **and Masaaki Nagata**[‡]
[†]Graduate School of Systems and Information Engineering, University of Tsukuba, Japan
[‡]NTT Communication Science Laboratories, NTT Corporation, Japan

Abstract

In this paper, we propose a multi-hop attention for the Transformer. It refines the attention for an output symbol by integrating that of each head, and consists of two hops. The first hop attention is the scaled dot-product attention which is the same attention mechanism used in the original Transformer. The second hop attention is a combination of multi-layer perceptron (MLP) attention and head gate, which efficiently increases the complexity of the model by adding dependencies between heads. We demonstrate that the translation accuracy of the proposed multi-hop attention outperforms the baseline Transformer significantly, +0.85 BLEU point for the IWSLT-2017 German-to-English task and +2.58 BLEU point for the WMT-2017 German-to-English task. We also find that the number of parameters required for a multi-hop attention is smaller than that for stacking another self-attention layer and the proposed model converges significantly faster than the original Transformer.

1 Introduction

Multi-hop attention was first proposed in end-to-end memory networks (Sukhbaatar et al., 2015) for machine comprehension. In this paper, we define a hop as a computational step which could be performed for an output symbol many times. By "multi-hop attention", we mean that some kind of attention is calculated many times for generating an output symbol. Previous multi-hop attention can be classified into "recurrent attention" (Sukhbaatar et al., 2015) and "hierarchical attention" (Libovický and Helcl, 2017). The former repeats the calculation of attention many times to refine the attention itself while the latter integrates attentions for multiple input information sources. The proposed multi-hop attention for the Transformer is different from previous recurrent attentions because the mechanism for the first hop attention and that for the second hop attention is different. It is also different from previous hierarchical attention because it is designed to integrate attentions from different heads for the same information source.

In neural machine translation, hierarchical attention (Bawden et al., 2018; Libovický and Helcl, 2017) can be thought of a multi-hop attention because it repeats attention calculation to integrate the information from multiple source encoders. On the other hand, in the Transformer (Vaswani et al., 2017), the state-of-the-art model for neural machine translation, feed-forward neural network (FFNN) integrates information from multiple heads. In this paper, we propose a multi-hop attention mechanism as a possible alternative to integrate information from multi-head attention in the Transformer.

We find that the proposed Transformer with multi-hop attention converges faster than the original Transformer. This is likely because all heads learn to influence each other, through a head gate mechanism, in the second hop attention (Figure 1). Recently, many Transformer-based pre-trained language models such as BERT have been proposed and take about a month for training. The speed at which the proposed model converges may be even more important than the fact that its accuracy is slightly better.

2 Multi-Hop Multi-Head Attention for the Transformer

2.1 Multi-Head Attention

One of the Transformer's major successes is multi-head attention, which allows each head to capture different features and achieve better results compared to a single-head case.

$$a^{(h)} = softmax(\frac{Q^{(h)}K^{(h)\mathrm{T}}}{\sqrt{d}})V^{(h)} \quad (1)$$

$$m = Concat(a^{(1)}, ..., a^{(h)})W_O \quad (2)$$

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 217–222
Florence, Italy, July 28 - August 2, 2019.

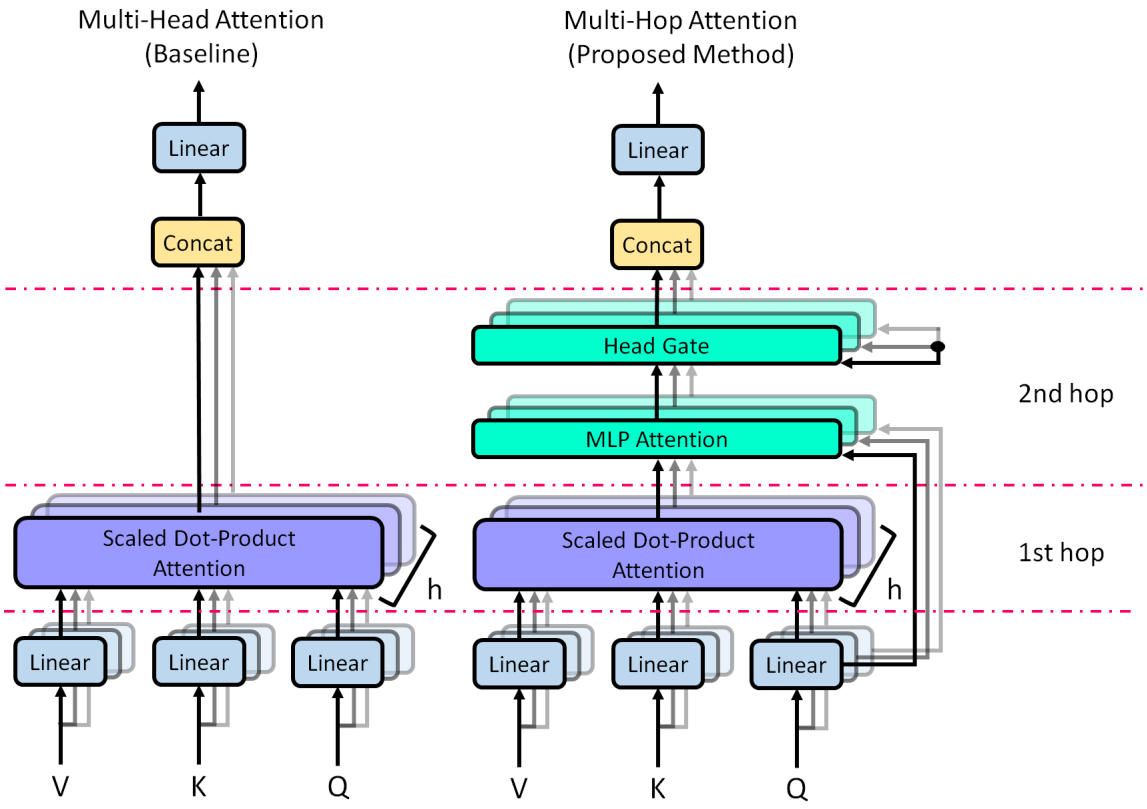


Figure 1: Multi-hop attention

Given the query Q, the key K, and the value V, they are divided into each head. Here, h $(= 1, \ldots, H)$ denotes the index of the head, where a is the output of scaled dot-product attention, W_O is a parameter for a linear transformation, and d is a scaling factor. Finally, the output of multi-head attention, m, is input to the next layer. The calculation of attention using scaled dot-product attention is defined as the first hop (Figure 1).

2.2 Multi-Hop Attention

In the original Transformer (Vaswani et al., 2017), information from each head is integrated by simple concatenation followed by a linear transformation. Attention is refined by stacking the combination of self-attention sub-layer and position-wise feed-forward neural network sub-layer. However, as layers are stacked, convergence becomes unstable. Consequently, there is a limit to the iterative approach by layering. Therefore, we propose a mechanism to repeat the calculation of attention based on a mechanism other than stacking layers.

The original Transformer is considered to consist of six single-hop attention layers. On the contrary, in the proposed method, some layers have a multi-hop (two-hop) attention. By experiments, we have established the appropriate position of the proposed multi-hop attention in the neural machine translation system. If the number of layers for encoders and decoders are six, then there are

Model	2nd hop	IWSLT2017 de→en	IWSLT2017 en→de
Baseline	-	33.46	27.21
Multi-hop	1	33.52	27.75†
Multi-hop	2	33.86†	27.98†
Multi-hop	3	33.74‡	27.98†
Multi-hop	4	**34.31**†	**28.08**†
Multi-hop	5	33.81†	27.81†
Multi-hop	6	33.83†	27.96†
Multi-hop	1,2	33.77‡	27.73†
Multi-hop	1,2,3	33.71‡	27.90†
Multi-hop	1,2,3,4	33.58	27.88†
Multi-hop	1,2,3,4,5	33.30	27.60†
Multi-hop	1,2,3,4,5,6	32.53	27.30
Multi-hop	2,3,4,5,6	32.80	27.54‡
Multi-hop	3,4,5,6	33.22	27.75†
Multi-hop	4,5,6	33.40	27.74†
Multi-hop	5,6	33.60	27.92†

†($p \leq 0.01$) and ‡($p \leq 0.05$) indicate that the proposed methods significantly outperform the Transformer baseline. The encoder and the decoder each had six layers, respectively.

Table 1: Best position for multi-hop

Model	IWSLT2017		WMT17	
	de→en	en→de	de→en	en→de
Baseline	33.46	27.21	21.33	18.15
Multi-hop	**34.31**†	**28.08**†	**23.91**†	**19.88**†

†($p \leq 0.01$) indicates that the proposed methods significantly outperform the Transformer baseline.

Table 2: Evaluation Result

Model	Layers	IWSLT2017	
		de→en	en→de
Vanilla	4	30.02	27.60
Multi-hop	4	30.09	27.63
Vanilla	5	33.80	28.00
Multi-hop	5	33.78	**28.15**
Vanilla	6	33.46	27.21
Multi-hop	6	**34.31**†	28.08†
Vanilla	7	31.80	26.58
Multi-hop	7	32.55†	27.36†

Table 3: Difference between 6-layer Transformer with multi-hop and 7-layer stacked vanilla Transformer

six self-attention layers in both the encoder and the decoder, respectively, and six source-to-target attention layers in the decoder.

The first hop attention of the multi-hop attention is equivalent to the calculation of scaled dot-product attention (Equation 1) in the original Transformer. The second hop attention consists of multi-layer perceptron (MLP) attention and head gate, as shown in Figure 1 and the following equations.

$$e_i^{(h)} = v_b^{\mathrm{T}} \tanh(W_b Q^{(h)} + U_b^{(h)} a_i^{(h)}) \quad (3)$$

$$\beta_i^{(h)} = \frac{\exp(e_i^{(h)})}{\sum_{n=1}^{N} \exp(e_i^{(h)})} \quad (4)$$

$$a_i'^{(h)} = \beta_i^{(h)} U_c^{(h)} a_i^{(h)} \quad (5)$$

First, MLP attention between the output of the first hop, $a_i^{(h)}$, and the query, Q, is calculated. Attention is considered as the calculation of a relationship between the query and the key/value. Therefore, in the second hop, attention is calculated again by using the output of the first hop, rather than the key/value.

Equations 4 and 5 are head gate in Figure 1. The head gate normalizes the attention score of each head to $\beta_i^{(h)}$, using the softmax function, where h ranges over all heads. In hierarchical attention (Bawden et al., 2018), the softmax function is used to select a single source from multiple sources. Here, the proposed head gate uses the softmax function to select a head from multiple heads. Finally, the head gate calculates new attention, $a_i'^{(h)}$, using the learnable parameters $U_c^{(h)}$, $\beta_i^{(h)}$, and $a_i^{(h)}$. The second hop MLP attention learns the optimal parameters for integration under the influence of the head gate. Although Vaswani et al. (2017) reported that dot-product attention is superior to MLP attention, we used MLP attention in the second hop of the proposed multi-hop attention because it can learn the dependence between heads by appropriately tuning the MLP parameters. We conclude that we can increase the expressive power of the network more efficiently by adding the second hop attention layer, rather than by stacking another single-hop multi-head attention layer.

Model	Layers	IWSLT2017	
		de→en	en→de
Vanilla	4	40,747K	41,882K
Multi-hop	4	40,763K	41,898K
Vanilla	5	48,103K	49,238K
Multi-hop	5	48,120K	49,254K
Vanilla	6	55,459K	56,594K
Multi-hop	6	55,492K	56.627K
Vanilla	7	62,816K	63,951K
Multi-hop	7	62,833K	63,967K

Table 4: Model Parameters

3 Experiment

3.1 Data

We used German-English parallel data obtained from the IWSLT2017 [1] and the WMT17 [2] shared tasks.

The IWSLT2017 training, validation, and test sets contain approximately 160K, 7.3K, and 6.7K sentence pairs, respectively. There are approximately 5.9M sentence pairs in the WMT17 training dataset. For the WMT17 corpus, we used newstest2013 as the validation set and newstest2014 and newstest2017 as the test sets.

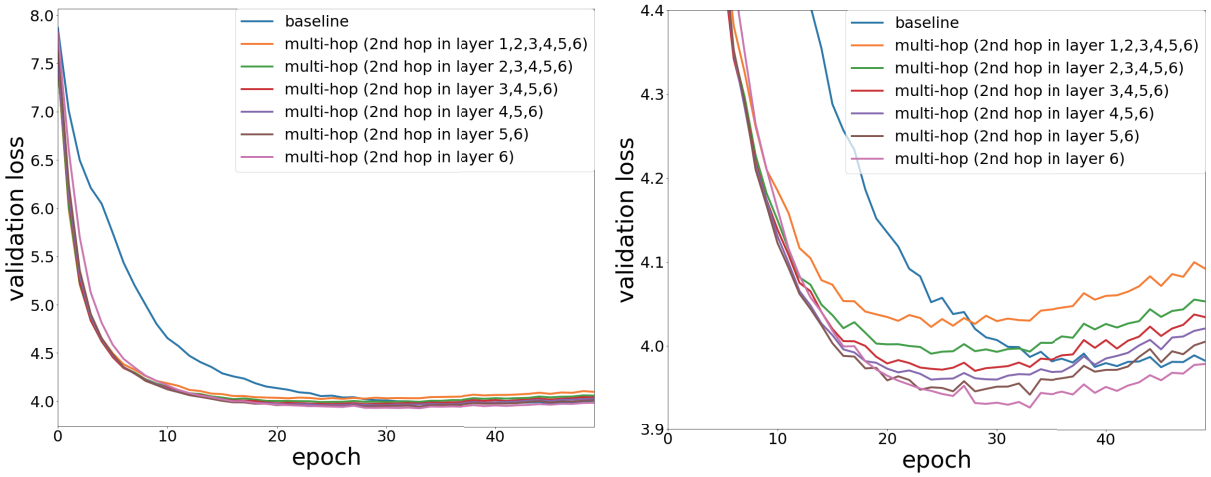
For tokenization, we used the subword-nmt tool (Sennrich et al., 2016) to set a vocabulary size of 32,000 for both German and English.

3.2 Experimental Setup

In our experiments, the baseline was the Transformer (Vaswani et al., 2017) model. We used

[1] https://sites.google.com/site/iwsltevaluation2017/

[2] http://www.statmt.org/wmt17/translation-task.html



(a) All learning curve view

(b) Enlarged view (loss 3.9 to 4.4)

Figure 2: Validation loss by each epoch for IWSLT2017 de-en - second hop in layer n to 6

fairseq (Gehring et al., 2017) [3] toolkit and the source code will be available at our github repository [4]. For training, we used the Adam optimizer with a learning rate of 0.0003. The embedding size was 512, the hidden size was 2048, and the number of heads was 8. The encoder and the decoder each had six layers. The number of tokens per batch was 2,000. The number of training epochs for IWSLT2017 and WMT17 were 50 and 10, respectively. In all experiments using the IWSLT2017, models were trained on an Nvidia GeForce RTX 2080 Ti GPU, while in all experiments using the WMT17, models were trained on an Nvidia Tesla P100 GPU.

3.3 Results

Results of the evaluation are presented in Tables 1 and 2. In Table 2, the proposed multi-hop attention is used only at the fourth layer in the encoder. In the evaluation of German-to-English translation for IWSLT2017, the proposed method achieved a BLEU score of 34.31, which indicates that it significantly outperforms the Transformer baseline, which returned a BLEU score of 33.46. For WMT17, the proposed method achieved a BLEU score of 23.91, indicating that it also significantly outperformed the Transformer baseline, which returned a BLEU score of 21.33.

In IWSLT2017 German-to-English and English-to-German translation tasks, various conditions were investigated, as shown in Table 1. The best models are shown in Figure 2.

The baseline training time was 1143.2s per epoch in IWSLT2017 German-to-English translation, and the training time for the proposed method is 1145.6s per epoch. We found that increasing the number of parameters did not affect training time.

4 Analysis

4.1 Difference between Multi-Hop and 7-layer Stacked Transformer

We compared the proposed method with the original Transformer. Table 3 shows the translation accuracies when the number of layers was changed from 4 to 7, encoder and decoder, respectively. Here, " Vanilla "refers to the original Transformer and " Multi-hop " refers to the proposed method where the multi-hop attention layer is used at the fourth layer in the encoder. As shown in Table 3, the 7-layer model BLEU score is lower than that of the 6-layer model. In the experiments, the number of parameters required by the 6- and 7-layer models was 55,459K, and 62,816K, respectively, and the number of parameters for the multi-hop method was 55,492K. The proposed method only increases the number of parameters by one percent compared to simply stacking one multi-head layer. Thus, it is evident that simply increasing the number of parameters and repeating the attention calculation doesn't necessarily improve performance. On the other hand, the proposed method does not improve the BLEU score when the number of layers is four and five. This is probably because the parameters of each head in the baseline Trans-

[3] https://github.com/pytorch/fairseq
[4] https://github.com/siida36/fairseq_mhda

Epoch	Baseline	Second hop					
		Layer 1,2,3,4,5,6	Layer 2,3,4,5,6	Layer 3,4,5,6	Layer 4,5,6	Layer 5,6	Layer 6
1	7.87	**7.49**	**7.49**	7.53	7.56	7.70	7.82
10	4.80	4.21	4.18	**4.17**	**4.17**	**4.17**	4.21
20	4.15	4.04	4.00	3.99	3.98	**3.97**	**3.97**
30	4.01	4.04	4.00	3.97	3.96	3.95	**3.93**
40	3.97	4.05	4.02	4.00	3.98	3.97	**3.94**
50	**3.98**	4.09	4.05	4.03	4.02	4.00	**3.98**

Table 5: Validation loss by epoch for IWSLT2017 de-en

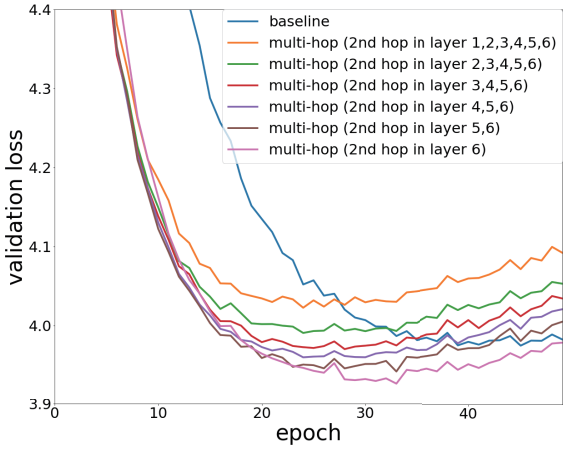


Figure 3: Validation loss by each epoch for IWSLT2017 de-en - second hop in only n layer

former are likely to converge properly when there are relatively few parameters. Another interpretation is that the normalization among heads forced by the proposed method works as noise.

As a conclusion, the proposed method demonstrates that appropriate connection can be obtained by recalculating attention in the layer where the head has a dependency.

Table 1 shows the effect of introducing second hop attention to various positions in the encoder. The second column shows the positions where the second hop attention is used. The best result was obtained when the second hop attention was used only for the fourth layer in the encoder. Performance decreased as the second hop attention was introduced to more layers, i.e., the worst result was obtained when using the second hop in all layers (second hop in layer 1,2,3,4,5,6). Further studies are needed to elucidate the relationship between performance and position of the second hop attention.

4.2 Effect on Learning Speed

Table 5 shows the validation loss of models for the IWSLT2017 German-to-English translation task with the second hop layers whose dropout rate is 30%. All models have 6 layers and the positions of the second hop layers have narrowed from all 6 layers to only 6th layers. It should be noted that, in the first epoch (row 1, Table 5), the model with the second hop in all layers has the lowest validation loss, while the baseline model has the highest validation loss.

Figure 2(a) shows the learning curve based on the same data shown in Table 5, It is apparent that the models with the second hop converge faster than the baseline model. Figure 2(b) is an enlarged view of Figure 2(a), focused on the lowest validation loss for different models. We find that the validation loss is lower when there are fewer second hop attentions.

Figure 3 shows the learning curves for the models with multi-hop attention used only once anywhere in layer 1 to 6. We find the model with second hop attention in layer 6 converges fastest. In terms of convergence, as opposed to accuracy, it seems appropriate to use second hop attention only in the last (6th) layer in the encoder.

5 Related Work

The mechanism of the proposed multi-hop attention for the Transformer was inspired by the hierarchical attention in multi-source sequence-to-sequence model (Libovický and Helcl, 2017). The term "multi-hop " is borrowed from the end-to-end memory network (Sukhbaatar et al., 2015) and the title "attention over heads" is inspired by Attention-over-Attention neural network (Cui et al., 2017), respectively.

Ahmed et al. (2018) proposed Weighted Transformer which replaces multi-head attention by multiple self-attention branches that learn to combine during the training process. They reported that it slightly outperformed the baseline Transformer (0.5 BLEU points on the WMT 2014 English-to-German translation task) and converges 15-40% faster. They linearly combined the multiple sources of attention, while we com-

bined multiple attention non-linearly using softmax function in the second hop.

It is well known that the Transformer is difficult to train (Popel and Bojar, 2018). As it has a large number of parameters, it takes time to converge and sometimes it does not do so at all without appropriate hyper parameter tuning. Considering the experimental results of our multi-hop attention experiments, and that of the Weight Transformer, an appropriate design of the network to combine multi-head attention could result in faster and more stable convergence of the Transformer. As the Transformer is used as a building block for the recently proposed pre-trained language models such as BERT (Devlin et al., 2019) which takes about a month for training, we think it is worthwhile to pursue this line of research including the proposed multi-hop attention.

Universal Transformer (Dehghani et al., 2019) can be thought of variable-depth recurrent attention. It obtained Turing-complete expressive power in exchange for a vast increase in the number of parameters and training time. As shown in Table 4, we have proposed an efficient method to increase the depth of recurrence in terms of the number of parameters and training time. Recently, Voita et al. (2019) and Michel et al. (2019) independently reported that only a certain subset of the heads plays an important role in the Transformer. They performed analyses by pruning heads from an already trained model, while we have proposed a method to assign weights to heads automatically. We assume our method (multi-hop attention or attention-over-heads) selects important heads in the early stage of training, which results in faster convergence than the original Transformer.

6 Conclusion

In this paper, we have proposed a multi-hop attention mechanism for a Transformer model in which all heads depend on each other repeatedly. We found that the proposed method significantly outperforms the original Transformer in accuracy and converges faster with little increase in the number of parameters. In future work, we would like to implement a multi-hop attention mechanism to the decoder side and investigate other language pairs.

References

K. Ahmed, N. S. Keskar, and R.Socher. 2018. Weighted transformer network for machine translation. *arXiv preprint arXiv:1711.02132*.

R. Bawden, R. Sennrich, A. Birch, and B. Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proc. NAACL-HLT*, pages 1304–1313.

Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proc. 55th ACL*, pages 593–602.

M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser. 2019. Universal transformers. In *Proc. 7th ICLR*.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, volume abs/1810.04805.

J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proc. 34th ICML*.

J. Libovický and J. Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proc. 55th ACL*, pages 196–202.

P. Michel, O. Levy, and G. Neubig. 2019. Are sixteen heads really better than one? *arXiv preprint arXiv:1905.10650*.

M. Popel and O. Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

R. Sennrich, B. Haddow, and A. Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. 54th ACL*, pages 1715–1725.

S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. 2015. End-to-end memory networks. In *Proc. 28th NIPS*, pages 2440–2448.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Proc. 30th NIPS*, pages 5998–6008.

E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.