Synchronous Syntactic Attention for Transformer Neural Machine Translation

Hiroyuki Deguchi
Nara Institute of Science and Technology
deguchi.hiroyuki.db0@is.naist.jp

Akihiro Tamura
Doshisha University
aktamura@mail.doshisha.ac.jp

Takashi Ninomiya
Ehime University
ninomiya@cs.ehime-u.ac.jp

Abstract

This paper proposes a novel attention mechanism for Transformer Neural Machine Translation, "Synchronous Syntactic Attention," inspired by synchronous dependency grammars. The mechanism synchronizes source-side and target-side syntactic self-attentions by minimizing the difference between target-side self-attentions and the source-side self-attentions mapped by the encoder-decoder attention matrix. The experiments show that the proposed method improves the translation performance on WMT14 En-De, WMT16 En-Ro, and ASPEC Ja-En (up to +0.38 points in BLEU).

1 Introduction

The Transformer Neural Machine Translation (NMT) model (Vaswani et al., 2017) has achieved state-of-the-art performance and become the focus of many NMT studies. One of its characteristics is the self-attention mechanism, which computes the strength of relationships between two words in a sentence. Transformer NMT has been improved by extending the self-attention mechanism to incorporate syntactic information (Wang et al., 2019b; Omote et al., 2019; Deguchi et al., 2019; Wang et al., 2019a; Bugliarello and Okazaki, 2020). In particular, Deguchi et al. (2019) and Wang et al. (2019a) have proposed dependency-based self-attentions, which are trained to attend to the syntactic parent for each token under constraints based on the dependency relations, for capturing sentence structures. Existing syntax-based NMT models, including their ones, use only monolingual syntactic information on either side or both.

By contrast, synchronous grammars such as synchronous context-free grammars and synchronous dependency grammars, which are defined in two languages and generate sentence structures aligned across them, have been introduced into many SMT models with the result of improving their translation performances (Jiang et al., 2009; Ding and Palmer, 2005; Chiang, 2005; Zhang et al., 2006). Figure 1 shows an example of the dependency structures of source and target language sentences and their alignments[1]. Inspired by synchronous dependency grammars, we aim to improve the performance of Transformer NMT by incorporating the main idea of the synchronous dependency grammars (i.e., synchronizing sentence structures across two languages). As far as we know, neither the synchronous dependency grammars themselves nor their basic idea has yet been incorporated into NMT.

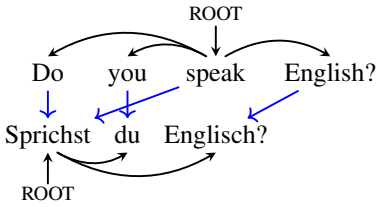


Figure 1: An example of dependency structures and alignments

This paper proposes a novel attention mechanism for Transformer NMT, called "Synchronous Syntactic Attention," which captures sentence structures aligned across two languages by the aligned self-attentions on the source- and target-side. The mechanism uses encoder-decoder attentions to map source-side syntactic self-attentions into a target language space based on Garg et al. (2019)'s observation that encoder-decoder attentions represent the alignments of source and target words. The mechanism is trained to maintain consistency between source- and target-side syntactic self-attentions according to an objective

[1] In this paper, an arrow is drawn from a head to its dependent.

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 348–355
August 5–6, 2021.

loss function that incorporates the difference between the target-side syntactic self-attentions and the mapped source-side syntactic self-attentions. We use *dependency-based self-attention* (Deguchi et al., 2019) as source- and target-side syntactic self-attentions.

2 Transformer NMT Model

The Transformer NMT model (Vaswani et al., 2017) is an encoder-decoder model composed of the encoder that encodes source tokens $\boldsymbol{f} = (f_1, f_2, \ldots, f_I)$ into hidden vectors and the decoder that generates target tokens $\boldsymbol{e} = (e_1, e_2, \ldots, e_J)$ from the outputs of the encoder. The encoder and decoder consist of N_{enc} encoder layers and N_{dec} decoder layers, respectively. Both the encoder layers and decoder layers are composed of multiple sub-layers, each of which includes a self-attention layer and a feed forward layer. The decoder layers additionally apply an encoder-decoder attention layer between the self-attention layer and the feed forward layer.

The self-attention and encoder-decoder attention are calculated by a multi-head attention mechanism. The multi-head attention $\mathrm{MHA}(Q, K, V)$ maps the d_{emb}-dimension embedding space into H subspaces of the d_k ($= \frac{d_{emb}}{H}$) dimension and calculates attention in each subspace as shown in Equations 1 to 3:

$$\mathrm{MHA}(Q, K, V) = [M_1; \ldots; M_H]W^M, \quad (1)$$

$$M_h = A_h V_h, A_h = \mathrm{softmax}\left(\frac{Q_h K_h^\top}{\sqrt{d_k}}\right), \quad (2)$$

$$Q_h = QW_h^Q, K_h = KW_h^K, V_h = VW_h^V, \quad (3)$$

where W_h^Q, W_h^K, $W_h^V \in \mathbb{R}^{d_{emb} \times d_k}$ and $W^M \in \mathbb{R}^{d_{emb} \times d_{emb}}$ are parameter matrices. In the self-attention, the previous layer's output is used as Q, K, and V. In the encoder-decoder attention, the previous layer's output is used as Q and the last encoder layer's output is used as K and V. Note that, in training, the decoder's self-attention masks future tokens.

3 Dependency-Based Self-Attention

This section describes *dependency-based self-attention (DBSA)* (Deguchi et al., 2019), which is the baseline of our syntactic self-attention. DBSA captures dependency structures by extending the multi-head self-attention of the l_{dep}-th layer of the encoder or decoder. Let h be one of head of the l_{dep}-th encoder layer's self-attention or the l_{dep}-th decoder layer's self attention. An attention weight matrix A_h, where each value indicates the dependency relationship between two words, is calculated by using the bi-affine operation in Equation 4:

$$A_h = \mathrm{softmax}\left(\frac{Q_h U K_h^\top}{\sqrt{d_k}}\right), U \in \mathbb{R}^{d_k \times d_k}. \quad (4)$$

In A_h, the probability of token q being the head of token t in a source/target sentence S (i.e., $P(q = \mathrm{head}(t)|S)$) is modeled as $A_h[t, q]$. Then, a weighted representation matrix M_h, which includes dependency relationships in the source sentence or target sentence, is obtained by multiplying A_h and V_h (i.e., $M_h = A_h V_h$). Finally, M_h is concatenated with the other heads and mapped to a d_{emb}-dimensional matrix. In the decoder-side DBSA, future information is masked to prevent attending to unpredicted tokens in inference.

The Transformer NMT model with DBSA learns translation and dependency parsing at the same time by minimizing the objective function $\mathcal{L} = \mathcal{L}_t + \lambda_{dep}\mathcal{L}_{dep}$, where $\mathcal{L}_t$ is the translation loss and $\mathcal{L}_{dep}$ is computed in Equation 5:

$$\mathcal{L}_{dep} = -\sum_{i=1}^{I} \log P(\mathrm{head}(f_i) \mid \boldsymbol{f}) - \sum_{j=1}^{J} \log P(\mathrm{head}(e_j) \mid \boldsymbol{e}). \quad (5)$$

$\lambda_{dep} > 0$ is a hyperparameter to control the influence of the dependency parsing loss $\mathcal{L}_{dep}$.

DBSA has been extended to deal with subword tokens. For details, see the original paper by Deguchi et al. (2019).

4 Proposed Method: Synchronous Syntactic Attention

This section proposes a novel attention mechanism for Transformer NMT, "Synchronous Syntactic Attention," which captures sentence structures aligned across source and target languages. A Transformer NMT model with the proposed attention is trained according to the objective function presented below as Equation 6:

$$\mathcal{L} = \mathcal{L}_t + \lambda_{dep}\mathcal{L}_{dep} + \lambda_{sync}\mathcal{L}_{sync}, \quad (6)$$

where $\mathcal{L}_{sync}$ is the loss to keep consistency between source-side and target-side syntactic self-

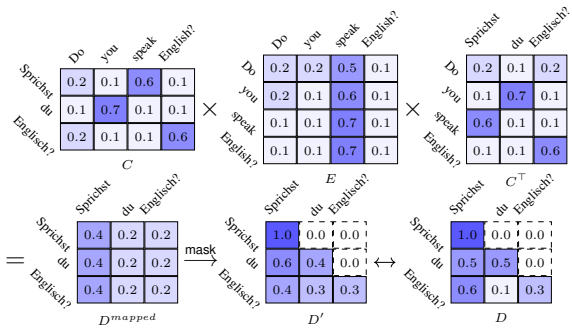


Figure 2: An example of synchronous syntactic attention

attention (i.e., DBSA) and λ_{sync} is a hyperparameter to control the influence of $\mathcal{L}_{sync}$. In particular, $\mathcal{L}_{sync}$ is the differences between the encoder's self-attention, which is mapped into target language space by the encoder-decoder attention, and the decoder's self-attention.

Let E and D be the attention matrix A_h of the l_{dep}-th encoder layer's syntactic self-attention and that of the l_{dep}-th decoder layer's syntactic self attention, respectively. The proposed method first maps E into the target language space by the encoder-decoder attention as shown by Equation 7:

$$D^{mapped} = CEC^\top, \tag{7}$$

where D^{mapped} is the mapped encoder's syntactic self attention matrix, and C is the encoder-decoder attention weight matrix of the l_{sync}-th decoder's layer. Then, D^{mapped} is masked to prevent attending to future tokens, and a softmax function is applied to the masked D^{mapped} as follows in Equation 8:

$$D' = \mathrm{softmax}(\mathrm{mask}(D^{mapped})). \tag{8}$$

Next, the proposed method computes the mean squared error between D' and D as $\mathcal{L}_{sync}$ as follows in Equation 9:

$$\mathcal{L}_{sync} = \sum_{t,q}(D'_{t,q} - D_{t,q})^2. \tag{9}$$

Figure 2 shows an example of the synchronous syntactic attention. The value in each cell indicates an attention score (i.e., an element of an attention weight matrix), and the darker cell represents a higher attention score. In all matrices, each row represents an attention distribution for each token (i.e., scores are normalized in a row direction). As can be seen in Figure 2, the English encoder's syntactic self-attentions E is mapped into the German encoder's syntactic self-attentions D' using the encoder-decoder attentions C and $C^\top$. Then, the loss between the German encoder's syntactic self-attentions D' and the German decoder's syntactic self-attentions D is measured. When calculating the loss, the values of the masked elements in D' and D, such as $D_{\text{Sprichst,du}}$ and $D_{\text{du,Englisch?}}$, are assigned to zero.

5 Experiments

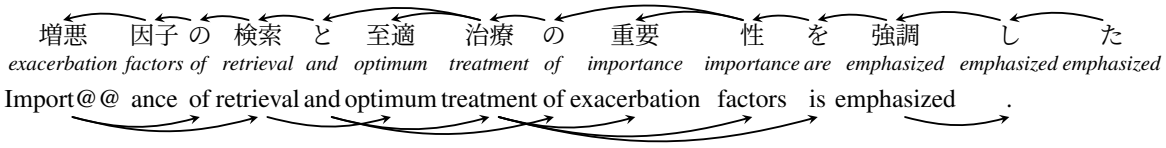
5.1 Setup

We compared the proposed model with a conventional Transformer NMT model and a Transformer NMT with DBSA (Transformer+DBSA), which do not synchronize between source- and target-side self attentions, to confirm the effectiveness of the proposed synchronous syntactic attention. The Transformer *base* model (Vaswani et al., 2017) was used as the baseline model.

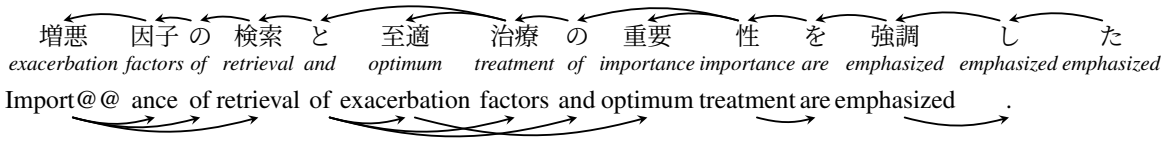
We evaluated translation performance in the WMT14 En-De translation task, WMT16 En-Ro translation task, and WAT ASPEC Ja-En translation task (Nakazawa et al., 2016). In ASPEC Ja-En, we used the first 1.5 million translation pairs of the training data in training. We used Moses Tokenizer to tokenize English, German, and Romanian sentences and KyTea (Neubig et al., 2011) to tokenize Japanese sentences. Byte Pair Encoding (BPE) was applied to create subword tokens. We used dependency structures generated by `Stanza` (Qi et al., 2020) for English, German, and Romanian sentences, and `EDA`[2] for Japanese sentences as the supervisions in the training of source- and target-side DBSA (i.e., calculation of $\mathcal{L}_{dep}$ in Transformer+DBSA and the proposed model). Note that `Stanza` and `EDA` are not used in testing. The details of the dataset and preprocessing are shown in the Appendix.

All models were trained for 100,000 updates. We used label smoothed cross entropy (Szegedy et al., 2016) as the $\mathcal{L}_t$ of the objective function and set label smoothing ϵ to 0.1. In the proposed model, the hyperparameter λ_{sync} was tuned for each development set and set to 0.5 for WMT14 En-De, 0.1 for WMT16 En-Ro, and 10.0 for ASPEC Ja-En. In all experiments, λ_{dep} and l_{dep} were set to 0.5 and 1, respectively. l_{sync} was set to 5 according to Garg et al. (2019)'s finding that the

[2] `http://www.ar.media.kyoto-u.ac.jp/tool/EDA`



(a) Dependency structures captured by DBSA's attentions



(b) Dependency structures captured by SyncAttn's attentions

Figure 3: Dependency structures of the examples in Figure 4

Model	WMT14 En→De	WMT16 En→Ro	ASPEC Ja→En
Transformer	27.23	23.83	28.94
DBSA	27.31	24.13	29.57
SyncAttn	**27.69**	**24.33**	**29.84**

Table 1: Experimental results (BLEU(%))

Input	増悪因子の検索と至適治療の重要性を強調した
DBSA	*Importance of **retrieval and optimum treatment of exacerbation factors** is emphasized.*
SyncAttn	*Importance of **retrieval of exacerbation factors and optimum treatment** are emphasized.*
Reference	*The importance of finding out exacerbation factors and optimum treatment are emphasized.*

Figure 4: Translation examples of DBSA and SyncAttn in the ASPEC Ja-En task

alignment performance of the encoder-decoder attention in the penultimate layer is the best among all layers. In decoding, we used beam search with length penalty and set the beam size to 4. The details of the hyperparameters are shown in the Appendix.

5.2 Results

Table 1 shows the experiment results. In the table, "DBSA" and "SyncAttn" indicate Transformer NMT with DBSA and Transformer NMT with the proposed synchronous syntactic attention, respectively. Translation performance was evaluated by BLEU (Papineni et al., 2002).

As Table 1 illustrates, the proposed model SyncAttn outperforms the baseline models Transformer and DBSA on all the tasks. In particular, SyncAttn improved by 0.38, 0.20, and 0.27 BLEU points in the WMT14 En-De, WMT16 En-Ro, ASPEC Ja-En tasks, respectively, compared to DBSA. These results demonstrate the effectiveness of our synchronous syntactic attention.

5.3 Case Study

This section compares translation examples of the baseline model DBSA and the proposed model SyncAttn to show the effectiveness of the synchronous syntactic attention. Figure 4 shows translation examples of the two models for the Ja-En task. The bold words are the differences between the translations by the two models. As can be seen in Figure 3, in both models, the encoder's self-attentions correctly find that "因子 (*factors*)" attends to "の (*of*)". However, DBSA does not correctly find the head of "factors" on the English side, while SyncAttn does. This is because SyncAttn synchronizes the source- and target-side dependency structures between "因子" and "factors" identified by the encoder-decoder attentions while DBSA does not. Figure 3 and 4 show that the correct analysis for the target-side dependency structures led to the correct translation.

6 Related Work

The main characteristic of Transformer NMT is attention mechanisms (i.e., self-attentions and encoder-decoder attentions). Some researches have analyzed and/or improved the attention mechanisms of Transformer NMT. For instance, Tang et al. (2018b) analyzed encoder-decoder attentions in terms of word sense disambiguation, and Tang et al. (2018a) analyzed self-attentions in terms of subject-verb agreement and word sense disambiguation. Raganato and Tiedemann (2018) and Voita et al. (2019) revealed the behaviors of attention heads in terms of dependency relations. Namely, Raganato and Tiedemann (2018) observed that specific attention heads of the en-

coder's self-attentions mark syntactic dependency relations. Voita et al. (2019) found that the confident heads play linguistically-interpretable roles like dependency relations. Garg et al. (2019) proposed a method for jointly learning to produce translations and alignments with a single Transformer model and showed that encoder-decoder attentions emulate word alignments. Based on their observations, our method maps the encoder's syntactic self-attentions into the target language space by using encoder-decoder attentions.

Shaw et al. (2018) extended a self-attention mechanism to encode the relative positions between two words in a sentence. Omote et al. (2019) and Wang et al. (2019b) proposed a self-attention mechanism to encode relative positions on source-side dependency trees.

Some researchers proposed syntax-aware self-attentions that are trained using dependency-based constraints. For instance, Wang et al. (2019a) and Bugliarello and Okazaki (2020) proposed source-side dependency-aware Transformer NMT. Wang et al. (2019a) created a constraint based on dependency relations between tokens to encoder self-attentions. Bugliarello and Okazaki (2020) also proposed *Parent-Scaled Self-Attention*, which multiplies an attention weight matrix by scores based on dependency relations. Deguchi et al. (2019) proposed *DBSA*, which is applicable to both the encoder's and decoder's self-attentions and is extended to subword units. We used DBSA to implement source- and target-side syntactic attentions in Transformer NMT. The main difference from the above-mentioned studies is that our work focuses on the incorporation of bilingual syntactic information into NMT.

Harada and Watanabe (2021) incorporated synchronous phrase structure grammar into NMT. Specifically, they proposed a syntactic NMT model that induces latent phrase structure and synchronizes the source- and target-side sentence structures. The difference with our model is that we synchronize dependency structures while they synchronize phrase structures.

7 Conclusions

In this paper, we proposed a novel attention mechanism for Transformer NMT, "Synchronous Syntactic Attention," which captures sentence structures aligned across source and target languages by aligned self-attention. The synchronous attention mechanism trains syntactic self-attentions (DBSA) under a constraint that minimizes the loss between encoder's and decoder's self attentions, where the encoder's self attentions are mapped into the target language space by encoder-decoder attentions. Since this method relies only on the constraint induced from the encoder's and decoder's self-attentions and encoder-decoder attentions, it does not require additional model parameters. The experiments show that the proposed method improves Transformer NMT's translation performance (up to a 0.38 BLEU point improvement).

Acknowledgments

The research results are achieved by "Research and Development of Deep Learning Technology for Advanced Multilingual Speech Translation," the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN. This work was partially supported by JSPS KAKENHI Grant Number JP20K19864 and JP21K12031.

References

Emanuele Bugliarello and Naoaki Okazaki. 2020. Enhancing machine translation with dependency-aware self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1618–1627, Online. Association for Computational Linguistics.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan. Association for Computational Linguistics.

Hiroyuki Deguchi, Akihiro Tamura, and Takashi Ninomiya. 2019. Dependency-based self-attention for transformer nmt. In *Proceedings of Recent Advances in Natural Language Processing*, pages 239–246.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 541–548, Ann Arbor, Michigan. Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align

and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Shintaro Harada and Taro Watanabe. 2021. Neural machine translation with synchronous latent phrase structure. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop (ACL-IJCNLP SRW 2021) (to appear)*.

Hongfei Jiang, Muyun Yang, Tiejun Zhao, Sheng Li, and Bo Wang. 2009. A statistical machine translation model based on a synthetic synchronous grammar. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 125–128, Suntec, Singapore. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *Proc. of LREC 2016*, pages 2204–2208.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Yutaro Omote, Akihiro Tamura, and Takashi Ninomiya. 2019. Dependency-based relative positional encoding for transformer nmt. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2019 (RANLP 2019)*, pages 854–861.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages.

Alessandro Raganato and Jörg Tiedemann. 2018. 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*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. 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, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018a. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272, Brussels, Belgium. Association for Computational Linguistics.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018b. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 26–35, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Chengyi Wang, Shuangzhi Wu, and Shujie Liu. 2019a. Source dependency-aware transformer

with supervised self-attention. *arXiv preprint arXiv:1909.02273*.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019b. Self-attention with structural position representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1403–1409, Hong Kong, China. Association for Computational Linguistics.

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

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 256–263, New York City, USA. Association for Computational Linguistics.

A Dataset and Preprocessing Details

We used Moses Tokenizer with the aggressive hyphen splitting option[3] for English, German, and Romanian sentences and KyTea for Japanese sentences. In English, German, and Romanian sentences, we used `normalize-punctuation.perl`, contained in the Moses toolkit, to normalize the characters. In WMT14 En-De, we also applied language identification filtering to the training data using `langid`[4] (Lui and Baldwin, 2012), keeping only the sentence pairs with correct languages on both sides (Ng et al., 2019). In ASPEC Ja-En, we used the first 1.5 million translation pairs of the training data in training. We trained Byte Pair Encoding (BPE) with 37,000 joint operations for WMT14 En-De and 40,000 joint operations for WMT16 En-Ro and trained BPE separately on the source and target sides with 16,000 merge operations for ASPEC Ja-En. We set the batch size to 25,000 tokens for WMT14 En-De, 6,000 tokens for WMT16 En-Ro, and 12,000 tokens for ASPEC Ja-En. Before applying BPE, we removed sentences longer than 100 words in all the training datasets and sentence pairs with a source/target length ratio exceeding 1.5 for WMT14 En-De and WMT16 En-Ro and 2.0 for ASPEC Ja-En.

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[4] https://github.com/saffsd/langid.c

	# Sentence pairs		
Dataset	Train	Dev	Test
WMT14 En→De	3,772,107	3,000	3,003
WMT16 En→Ro	599,208	1,999	1,999
ASPEC Ja→En	1,428,181	1,790	1,812

Table 2: Statistics of evaluation dataset

Table 2 shows the number of parallel sentence pairs in the training, development, and test sets.

B Model and Training Details

We used the Transformer *base* model (Vaswani et al., 2017) as the baseline model. We used the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The learning rate was warmed up over the first 4,000 steps to a peak value of 7e-4, and then it was decreased proportionally to the inverse square root of the step number (Vaswani et al., 2017). All models were trained for 100,000 updates. The dropout probability was set to 0.1. We used label smoothed cross entropy (Szegedy et al., 2016) as the $\mathcal{L}_t$ of the objective function and set label smoothing ϵ to 0.1. In all experiments, λ_{dep} was set to 0.5, the l_{dep}-th layer that captures source or target side's sentence structures was set to the 1st (bottom) layer, and the encoder-decoder attention for mapping the encoder's self-attention was obtained from the 5th layer (i.e., l_{sync}=5) according to Garg et al. (2019)'s finding that the alignment performance of the encoder-decoder attention in the penultimate layer is the best among all layers. In decoding, we used beam search with a beam size of 4 and length penalty $\alpha = 0.6$ (Wu et al., 2016).

We performed all the training on 2 V100 GPUs for WMT14 En-De, and a single V100 GPU for WMT16 En-Ro and ASPEC Ja-En. For all the models, training took about 7 hours for WMT14 En-De, about 3 hours for WMT16 En-Ro, and about 4 hours for ASPEC Ja-En. The number of model parameters of all models is about 64M for WMT14 En-De and WMT16 En-Ro, and about 72M for ASPEC Ja-En. In WMT14 En-De and WMT16 En-Ro, the encoder-side embedding layer and the decoder-side embedding layer are shared.

C Hyperparameter Search

In the proposed model, the hyperparameter λ_{sync} was tuned on each development set. We tuned λ_{sync} by trying different $\lambda_{sync} \in \{0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0\}$.

D Evaluation Details

In all experiments, translation performance was evaluated by BLEU (Papineni et al., 2002). As for the ASPEC Ja-En task, we followed the WAT Automatic Evaluation Systems[5].

[5] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/index.html#automatic_evaluation_systems.html