Dependency-Based Self-Attention for Transformer NMT

Hiroyuki Deguchi, Akihiro Tamura, Takashi Ninomiya
Ehime University
{deguchi@ai., tamura@, ninomiya@}cs.ehime-u.ac.jp

Abstract

In this paper, we propose a new Transformer neural machine translation (NMT) model that incorporates dependency relations into self-attention on both source and target sides, dependency-based self-attention. The dependency-based self-attention is trained to attend to the modifiee for each token under constraints based on the dependency relations, inspired by linguistically-informed self-attention (LISA). While LISA was originally designed for Transformer encoder for semantic role labeling, this paper extends LISA to Transformer NMT by masking future information on words in the decoder-side dependency-based self-attention. Additionally, our dependency-based self-attention operates at subword units created by byte pair encoding. Experiments demonstrate that our model achieved a 1.0 point gain in BLEU over the baseline model on the WAT'18 Asian Scientific Paper Excerpt Corpus Japanese-to-English translation task.

1 Introduction

In the field of machine translation (MT), the Transformer model (Vaswani et al., 2017) has outperformed recurrent neural network (RNN)-based models (Sutskever et al., 2014) and convolutional neural network (CNN)-based models (Gehring et al., 2017) on many translation tasks, and thus has garnered attention from MT researchers. The Transformer model computes the strength of a relationship between two words in a sentence by means of a self-attention mechanism, which has contributed to the performance improvement in not only MT but also various NLP tasks such as language modeling and semantic role labeling (SRL).

The performance of MT, including statistical machine translation and RNN-based neural machine translation (NMT), has been improved by incorporating sentence structures (Lin, 2004; Chen et al., 2017; Eriguchi et al., 2017; Wu et al., 2018). In addition, Strubell et al. (2018) have improved a Transformer-based SRL model by incorporating dependency structures of sentences into self-attention, which is called linguistically-informed self-attention (LISA). In LISA, one attention head of a multi-head self-attention is trained with constraints based on dependency relations to attend to syntactic parents for each token.

In the present work, we aim to improve translation performance by utilizing dependency relations in Transformer NMT. To this end, we propose a Transformer NMT model that incorporates dependency relations into self-attention on both source and target sides. Specifically, in training, a part of self-attention is learned with constraints based on dependency relations of source or target sentences to attend to a modifiee for each token, and, in decoding, the proposed model translates a sentence in consideration of dependency relations in both the source and target sides, which are captured by our self-attention mechanisms. Hereafter, the proposed self-attention is called dependency-based self-attention. Note that the dependency-based self-attention is inspired by LISA, but the straightforward adaptation of LISA, which is proposed for Transformer encoder, does not work well for NMT because a target sentence is not fully revealed in inference. Therefore, the proposed model masks future information on words in the decoder-side dependency-based self-attention to prevent from attending to unpredicted subsequent tokens.

Recent NMT models treat a sentence as a sub-

Proceedings of Recent Advances in Natural Language Processing, pages 239–246,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_028

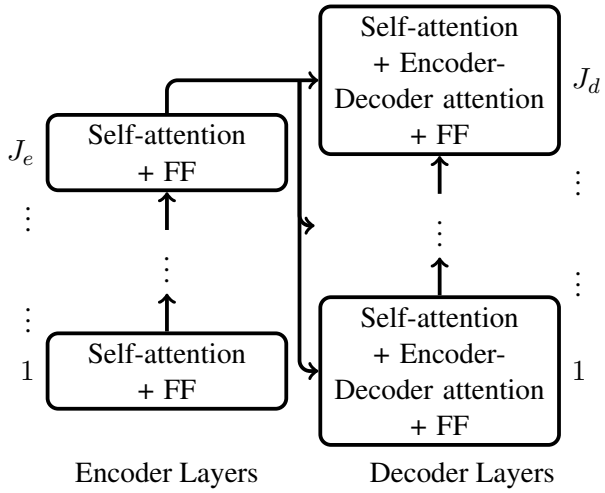


Figure 1: Transformer model.

word sequence rather than a word sequence to address the translation of out-of-vocabulary words (Sennrich et al., 2016). Therefore, we extend dependency-based self-attention to operate at subword units created by byte pair encoding (BPE) rather than word-units.

Our experiments demonstrate that the proposed Transformer NMT model performs 1.0 BLEU points higher than the baseline Transformer NMT model, which does not incorporate dependency structures, on the WAT'18 Asian Scientific Paper Excerpt Corpus (ASPEC) Japanese-to-English translation task. The experiments also demonstrate the effectiveness of each of our proposals, namely, encoder-side dependency-based self-attention, decoder-side dependency-based self-attention, and extension for BPE.

2 Transformer NMT

We provide here an overview of the Transformer NMT model (Vaswani et al., 2017), which is the basis of our proposed model. The outline of the Transformer NMT model is shown in Fig. 1.

The Transformer NMT model is an encoder-decoder model that has a self-attention mechanism. The encoder maps an input sequence of symbol representations (i.e., a source sentence) $X = (x_1, x_2, \ldots, x_{n_{enc}})^T$ to an intermediate vector. Then, the decoder generates an output sequence (i.e., a target sentence) $Y = (y_1, y_2, \ldots, y_{n_{dec}})^T$, given the intermediate vector. The encoder and the decoder are composed of a stack of J_e encoder layers and of J_d decoder layers, respectively.

Because the Transformer model does not include recurrent or convolutional structures, it encodes word positional information as sinusoidal positional encodings:

$$P_{(pos,2i)} = sin(pos/10000^{2i/d}), \quad (1)$$
$$P_{(pos,2i+1)} = cos(pos/10000^{2i/d}), \quad (2)$$

where pos is the position, i is the dimension, and d is the dimension of the intermediate representation. At the first layers of the encoder and decoder, the positional encodings calculated by Equations (1) and (2) are added to the input embeddings.

The j-th encoder layer's output $S_{enc}^{(j)}$ is generated by a self-attention layer $SelfAttn()$ and a position-wise fully connected feed forward network layer $FFN()$ as follows:

$$H_{enc}^{(j)} = LN(S_{enc}^{(j-1)} + SelfAttn(S_{enc}^{(j-1)})), \quad (3)$$
$$S_{enc}^{(j)} = LN(H_{enc}^{(j)} + FFN(H_{enc}^{(j)})), \quad (4)$$

where $S_{enc}^{(0)}$ is the input of the encoder, $H_{enc}^{(j)}$ is the output of the j-th encoder's self-attention, and $LN()$ is layer normalization (Lei Ba et al., 2016). The j-th decoder layer's output $S_{dec}^{(j)}$ is generated by an encoder-decoder attention layer $EncDecAttn()$ in addition to the two sublayers of the encoder (i.e., $SelfAttn()$ and $FFN()$) as follows:

$$H_{dec}^{(j)} = LN(S_{dec}^{(j-1)} + SelfAttn(S_{dec}^{(j-1)})), \quad (5)$$
$$G_{dec}^{(j)} = LN(H_{dec}^{(j)} + EncDecAttn(H_{dec}^{(j)})), \quad (6)$$
$$S_{dec}^{(j)} = LN(G_{dec}^{(j)} + FFN(H_{dec}^{(j)})), \quad (7)$$

where $S_{dec}^{(0)}$ is the input of the decoder, $H_{dec}^{(j)}$ is the output of the j-th decoder's self-attention, and $G_{dec}^{(j)}$ is the output of the j-th decoder's encoder-decoder attention.

The last decoder layer's output $S_{dec}^{(J_d)}$ is linearly mapped to a V-dimensional matrix, where V is the output vocabulary size. Then, the output sequence Y is generated based on $P(Y \mid X)$, which is calculated by applying the softmax function to the V-dimensional matrix.

Self-attention computes the strength of the relationship between two words in the same sentence (i.e., between two source words or between two target words), and encoder-decoder attention computes the strength of the relationship between a source word and a target word.

Both the self-attention and encoder-decoder attention are implemented with multi-head attention, which projects the embedding space into n_{head} subspaces of the $d_{head} = d/n_{head}$ dimension and calculates attention in each subspace. In the j-th layer's self-attention, the previous layer's output $S^{(j-1)} \in \mathbb{R}^{n \times d}$ is linearly mapped to three d_{head}-dimensional subspaces, $Q_h^{(j)}$, $K_h^{(j)}$, and $V_h^{(j)}$, using parameter matrices $W_h^{Q^{(j)}} \in \mathbb{R}^{d \times d_{head}}$, $W_h^{K^{(j)}} \in \mathbb{R}^{d \times d_{head}}$, and $W_h^{V^{(j)}} \in \mathbb{R}^{d \times d_{head}}$, where n is the length of the input sequence and $1 \leq h \leq n_{head}$[1]. In the j-th decoder layer's encoder-decoder attention, the previous layer's output $S_{dec}^{(j-1)}$ is mapped to $Q_h^{(j)}$, and the last encoder layer's output $S_{enc}^{(J_e)}$ is mapped to $K_h^{(j)}$ and $V_h^{(j)}$.

Then, an attention weight matrix, where each value represents the strength of the relationship between two words, is calculated on each subspace as follows:

$$A_h^{(j)} = softmax(d_{head}^{-0.5} Q_h^{(j)} K_h^{(j)^T}). \quad (8)$$

By multiplying $A_h^{(j)}$ and $V_h^{(j)}$, a weighted representation matrix $M_h^{(j)}$ is obtained:

$$M_h^{(j)} = A_h^{(j)} V_h^{(j)}. \quad (9)$$

$M_h^{(j)}$ in self-attention includes the strengths of the relationships with all words in the same sentence for each source or target word, and $M_h^{(j)}$ in encoder-decoder attention includes the strengths of the relationships with all source words for each target word.

Finally, the concatenation of all $M_h^{(j)}$ (i.e., $M_{1,2,\ldots,n_{head}}^{(j)}$) is mapped to a d-dimensional matrix $M^{(j)}$ as follows:

$$M^{(j)} = W^{M^{(j)}} [M_1^{(j)}; \ldots; M_{n_{head}}^{(j)}], \quad (10)$$

where $W^{M^{(j)}} \in \mathbb{R}^{d \times d}$ is a parameter matrix.

Note that, in training, the decoder's self-attention masks future words so as to ensure that the attentions of a target word do not rely on unpredicted words in inference.

3 Proposed Method

Figure 2 shows the outline of the proposed model. The proposed model incorporates dependency relations into self-attention on both source and target sides, *dependency-based self-attention*. In particular, it parses the dependency structures of source sentences and target sentences by one attention head of the p_e-th encoder layer's multi-head self-attention and one of the p_d-th decoder layer's multi-head self-attention, respectively, and translates a sentence based on the source-side and target-side dependency structures. We use the deep bi-affine parser (Dozat and Manning, 2016) as a model for dependency parsing in the dependency-based self-attention according to LISA. There are two inherent differences between LISA and our dependency-based self-attention: (i) our decoder-side dependency-based self-attention masks future information on words, and (ii) our dependency-based self-attention operates at subword units created by byte pair encoding rather than word-units.

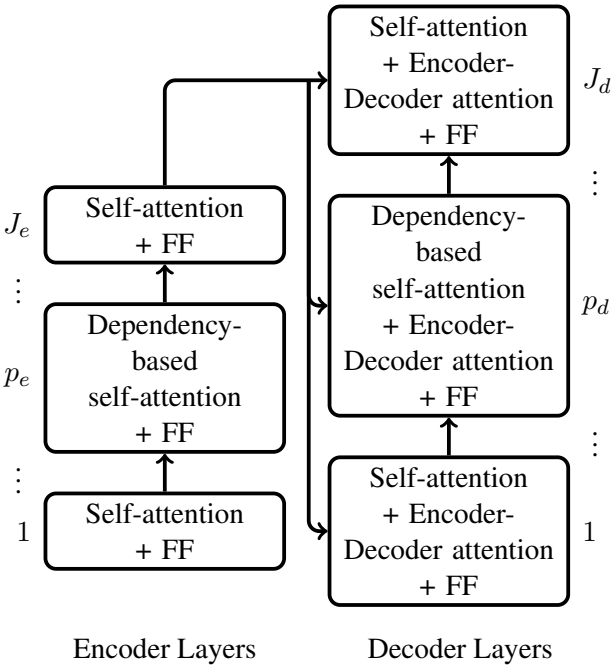


Figure 2: Proposed model.

3.1 Dependency-Based Self-Attention

The dependency-based self-attention parses dependency structures by extending the multi-head self-attention of the p-th layer of the encoder or decoder[2]. First, the p-th self-attention layer maps the previous layer's output $S^{(p-1)}$ of d-dimension to d_{head}-dimensional subspaces of multi-head at-

[1] $S^{(j)}$ indicates $S_{enc}^{(j)}$ for the encoder and $S_{dec}^{(j)}$ for the decoder.

[2] p indicates p_e for the encoder and p_j for the decoder.

tention as follows:

$$Q_{parse} = S^{(p-1)} W^{Q_{parse}}, \quad (11)$$
$$K_{parse} = S^{(p-1)} W^{K_{parse}}, \quad (12)$$
$$V_{parse} = S^{(p-1)} W^{V_{parse}}, \quad (13)$$

where $W^{Q_{parse}}$, $W^{K_{parse}}$, and $W^{V_{parse}}$ are $d \times d_{head}$ weight matrices. Next, an attention weight matrix A_{parse}, where each value indicates the dependency relationship between two words, is calculated by using the bi-affine operation as follows:

$$A_{parse} = softmax(Q_{parse} U^{(1)} K_{parse}^T + Q_{parse} U^{(2)}), \quad (14)$$

where $U^{(1)} \in \mathbb{R}^{d_{head} \times d_{head}}$, $U^{(2)} = \overbrace{(\mathbf{u} \ldots \mathbf{u})}^{n}$, and $\mathbf{u} \in \mathbb{R}^{d_{head}}$ are the parameters. In A_{parse}, the probability of token q being the head of token t (i.e., t modifying q) is modeled as $A_{parse}[t, q]$:

$$P(q = head(t) \mid X) = A_{parse}[t, q], \quad (15)$$

where X is a source sentence or a target sentence, and the root token is defined as having a self-loop (i.e., $q = head(t) = ROOT$). Then, a weighted representation matrix M_{parse}, which includes dependency relationships in the source sentence or target sentence, is obtained by multiplying A_{parse} and V_{parse}:

$$M_{parse} = A_{parse} V_{parse}. \quad (16)$$

Finally, after one attention head (e.g., $M^{(p)}_{n_{head}}$) is replaced with M_{parse}, the concatenation of all $M^{(p)}_h$ (i.e., M_{parse} and $M^{(p)}_{1,2,\ldots,n_{head}-1}$) is mapped to a d-dimensional matrix $M^{(p)}$ like the conventional multi-head attention:

$$M^{(p)} = W^{M^{(p)}} [M_{parse}; M^{(p)}_1; \ldots; M^{(p)}_{n_{head}-1}], \quad (17)$$

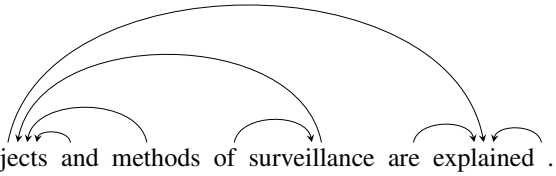
where $W^{M^{(p)}} \in \mathbb{R}^{d \times d}$ is a parameter matrix.

As can be seen in Equation (17), in the dependency-based self-attention, dependency relations are identified by one attention head M_{parse} of the p-th layer's multi-head attention.

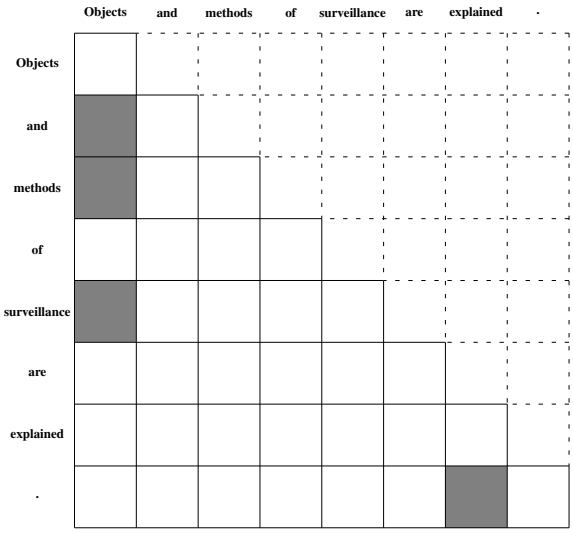
3.2 Objective Function

Our model learns translation and dependency parsing at the same time by minimizing the following objective function:

$$e^{tokens} + \lambda_{enc} e^{parse}_{enc} + \lambda_{dec} e^{parse}_{dec}, \quad (18)$$



(a) Dependency relationships.



(b) Attention matrix representing supervisions.

Figure 3: Decoder side masked dependency-based self-attention.

where e^{tokens} is the error of translation, and e^{parse}_{enc} and e^{parse}_{dec} are the errors of dependency parsing in the encoder and the decoder, respectively. $\lambda_{enc} > 0$ and $\lambda_{dec} > 0$ are hyper-parameters to control the influence of dependency parsing errors in the encoder and the decoder, respectively. e^{tokens} is calculated by label smoothed cross entropy (Szegedy et al., 2016), and e^{parse}_{enc} and e^{parse}_{dec} are calculated by cross entropy.

Note that, in the training of the decoder-side dependency-based self-attention, future information is masked to prevent attending to unpredicted tokens in inference. An example of training data for the decoder-side dependency-based self-attention is provided in Figure 3, where (a) is an example of dependency structures[3] and (b) shows the attention matrix representing the supervisions from (a). In (b), a dark cell indicates a dependency relation and a dotted cell means a masked

[3] In this paper, an arrow is drawn from a modifier to its modifiee. For example, the arrow drawn from "Objects" to "explained" indicates that "Objects" modifies "explained" (i.e., "explained" = $head$("Objects")).

	sentence pairs
train	1,198,149
dev	1,790
test	1,812

Table 1: Statistics of the ASPEC data.

Model	BLEU
Trans.	27.29
Trans. + DBSA(Enc)	28.05
Trans. + DBSA(Dec)	27.86
Trans. + DBSA(Enc) + DBSA(Dec)	**28.29**

Table 2: Translation performance.

element. As shown, future information on each word is masked. For example, the dependency relation from "are" to "explained" is masked.

3.3 Subword Dependency-Based Self-Attention

Recent NMT models have improved the translation performance by treating a sentence as a subword sequence rather than a word sequence. Therefore, we extend dependency-based self-attention to work for subword sequences. In our subword dependency-based self-attention, a sentence is divided into a subword sequence by BPE (Sennrich et al., 2016). When a word is divided into multiple subwords, the modifiee (i.e., the head) of the rightmost subword is set to the modifiee of the original word and the modifiee of each subword other than the rightmost one is set to the right adjacent subword.

Figure 4 shows an example of subword-level dependency relations, where "@@" is a subword segmentation symbol. "Fingerprint" is divided into the three subwords: "Fing@@", "er@@", and "print". When the head of the word "Fingerprint" is "input" in the original word-level sentence, the heads of the three subwords are determined as follows: "er@@" = head("Fing@@"), "print" = head("er@@"), and "input" = head("print").

4 Experiments

4.1 Experiment Settings

In our experiments, we compared the proposed model with a conventional Transformer NMT model, which does not incorporate dependency structures, to confirm the effectiveness of the proposed model. We stacked six layers for each encoder and decoder and set $n_{head} = 8$ and $d = 512$. For the proposed model, we incorporated dependency-based self-attention into the fourth layer in both the encoder and the decoder (i.e., $p_e = p_d = 4$).

We evaluated translation performance on the WAT'18 ASPEC (Nakazawa et al., 2016) Japanese-to-English translation task. We tokenized each Japanese sentence with $KyTea$ (Neubig et al., 2011) and preprocessed according to the recommendations from WAT'18[4]. We used the vocabulary of 100K subword tokens based on BPE for both languages and used the first 1.5 million translation pairs as the training data. In the training, long sentences with over 250 subword-tokens were filtered out. Table 1 shows the statistics of our experiment data.

We used Japanese dependency structures generated by EDA[5] and English dependency structures generated by Stanford Dependencies[6] in the training of the source-side dependency-based self-attention and the target-side dependency-based self-attention, respectively. Note that Stanford Dependencies and EDA are not used in the testing.

4.2 Training Details

We trained each model using Adam (Kingma and Ba, 2014), where the learning rate and hyperparameter settings are set following Vaswani et al. (2017). For the objective function, we set ϵ_{ls} (Szegedy et al., 2016) in label smoothing to 0.1 and both the hyperparameters λ_{enc} and λ_{dec} to 1.0. We set the mini-batch size to 224 and the number of epochs to 20. We chose the model that achieved the best BLEU score on the development set and evaluated the sentences generated from the test set using beam search with a beam size of 4 and length penalty $\alpha = 0.6$ (Wu et al., 2016).

4.3 Experiment Results

Table 2 lists the experiment results. Translation performance is measured by BLEU (Papineni et al., 2002). In Table 2, DBSA denotes our dependency-based self-attention. As shown, our proposed model

[4] http://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2018/baseline/dataPreparationJE.html

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

[6] https://nlp.stanford.edu/software/stanford-dependencies.html

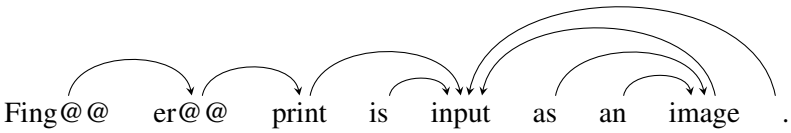


Figure 4: Subword-level dependency relationships.

Model	BPE	BLEU
Trans.	w/o	26.39
Trans.	w/	27.29
Trans. + DBSA(Enc) + DBSA(Dec)	w/o	26.62
Trans. + DBSA(Enc) + DBSA(Dec)	w/	**28.29**

Table 3: Effectiveness of subword.

"Trans.+DBSA(Enc)+DBSA(Dec)" performed significantly better than the baseline model "Trans.", which demonstrates the effectiveness of our dependency-based self-attention. Table 2 also shows that using either the encoder-side dependency-based self-attention or the decoder-side dependency-based self-attention improves translation performance, and using them in combination achieves further improvements.

5 Discussion

To determine the effectiveness of our extension to utilize subwords, we evaluated the models without BPE, where each sentence is treated as a word sequence. In the models without BPE, words that appeared fewer than five times in the training data were replaced with the special token "<UNK>". Table 3 lists the results. As shown, BPE improves the performance of both the baseline and the proposed model, which demonstrates the effectiveness of the subword dependency-based self-attention. Table 3 also shows that the proposed model outperforms the baseline model when BPE is not used. This strengthens the usefulness of our dependency-based self-attention.

6 Related Work

NMT models have been improved by incorporating source-side dependency relations (Chen et al., 2017), or target-side dependency relations (Eriguchi et al., 2017), or both (Wu et al., 2018).

Chen et al. (2017) have proposed SDRNMT, which computes dependency-based context vectors from source-side dependency trees by CNN and then uses the representations in the encoder of an RNN-based NMT model.

Eriguchi et al. (2017) have proposed NMT+RNNG, which combines the RNN-based dependency parser, RNNG (Dyer et al., 2016), and the decoder of an RNN-based NMT model.

Wu et al. (2018) have proposed a syntax-aware encoder, which encodes two extra sequences linearized from source-side dependency trees in addition to word sequences, and have incorporated Action RNN, which implements a shift-reduce transition-based dependency parsing by predicting action sequences, into the decoder. Their method has been applied to an RNN-based NMT model and a Transformer NMT model.

As far as we know, except for Wu et al. (2018), existing dependency-based NMT models have been based on RNN-based NMT. Although Wu et al. (2018) used dependency relations in Transformer NMT, they did not modify the Transformer model itself. In contrast, we have improved a Transformer NMT model to explicitly incorporate dependency relations (i.e., dependency-based self-attention). In addition, while Wu et al. (2018) need a parser for constructing source-side dependency structures in inference, our proposed method does not require an external parser in inference because the learned dependency-based self-attention of the encoder finds dependency relations.

7 Conclusion

In this paper, we have proposed a method to incorporate dependency relations on both source and target sides into Transformer NMT through

dependency-based self-attention. Our decoder-side dependency-based self-attention masks future information to avoid conflicts between training and inference. In addition, our dependency-based self-attention is extended to work well for subword sequences. Experimental results showed that the proposed model achieved a 1.0 point gain in BLEU over the baseline Transformer model on the WAT'18 ASPEC Japanese-English translation task. In future work, we will explore the effectiveness of our proposed method for language pairs other than Japanese-to-English.

Acknowledgement

The research results have been 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 JP18K18110.

References

Kehai Chen, Rui Wang, Masao Utiyama, Lemao Liu, Akihiro Tamura, Eiichiro Sumita, and Tiejun Zhao. 2017. Neural machine translation with source dependency representation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 2846–2852.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734* .

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 199–209.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. pages 72–78.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pages 1243–1252.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* .

Dekang Lin. 2004. A path-based transfer model for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*. pages 625–630.

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*.

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*. Association for Computational Linguistics, pages 529–533.

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*. ACL '02, pages 311–318.

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)*. Association for Computational Linguistics, Berlin, Germany, pages 1715–1725.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proc. of EMNLP 2018*.

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*, Curran Associates, Inc., pages 3104–3112.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *The IEEE Conference on CVPR*.

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

S. Wu, D. Zhang, Z. Zhang, N. Yang, M. Li, and M. Zhou. 2018. Dependency-to-dependency neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26(11):2132–2141.

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* .