Enhancing Machine Translation with Dependency-Aware Self-Attention

Emanuele Bugliarello[*]
University of Copenhagen
emanuele@di.ku.dk

Naoaki Okazaki
Tokyo Institute of Technology
okazaki@c.titech.ac.jp

Abstract

Most neural machine translation models only rely on pairs of parallel sentences, assuming syntactic information is automatically learned by an attention mechanism. In this work, we investigate different approaches to incorporate syntactic knowledge in the Transformer model and also propose a novel, parameter-free, dependency-aware self-attention mechanism that improves its translation quality, especially for long sentences and in low-resource scenarios. We show the efficacy of each approach on WMT English↔German and English→Turkish, and WAT English→Japanese translation tasks.

1 Introduction

Research in neural machine translation (NMT) has mostly exploited corpora consisting of pairs of parallel sentences, with the assumption that a model can automatically learn prior linguistic knowledge via an attention mechanism (Luong et al., 2015). However, Shi et al. (2006) found that these models still fail to capture deep structural details, and several studies (Sennrich and Haddow, 2016; Eriguchi et al., 2017; Chen et al., 2017, 2018) have shown that syntactic information has the potential to improve these models. Nevertheless, the majority of syntax-aware NMT models are based on recurrent neural networks (RNNs; Elman 1990), with only a few recent studies that have investigated methods for the Transformer model (Vaswani et al., 2017).

Wu et al. (2018) evaluated an approach to incorporate syntax in NMT with a Transformer model, which not only required three encoders and two decoders, but also target-side dependency relations (precluding its use to low-resource target languages). Zhang et al. (2019) integrate source-side syntax by concatenating the intermediate representations of a dependency parser to word embeddings. In contrast to ours, this approach does not allow to learn sub-word units at the source side, requiring a larger vocabulary to minimize out-of-vocabulary words. Saunders et al. (2018) interleave words with syntax representations which results in longer sequences – requiring gradient accumulation for effective training – while only leading to $+0.5$ BLEU on WAT Ja-En when using ensembles of Transformers. Finally, Currey and Heafield (2019) propose two simple data augmentation techniques to incorporate source-side syntax: one that works well on low-resource data, and one that achieves a high score on a large-scale task. Our approach, on the other hand, performs equally well in both settings.

While these studies improve the translation quality of the Transformer, they do not exploit its properties. In response, we propose to explicitly enhance the its self-attention mechanism (a core component of this architecture) to include syntactic information without compromising its flexibility. Recent studies have, in fact, shown that self-attention networks benefit from modeling local contexts by reducing the dispersion of the attention distribution (Shaw et al., 2018; Yang et al., 2018, 2019), and that they might not capture the inherent syntactic structure of languages as well as recurrent models, especially in low-resource settings (Tran et al., 2018; Tang et al., 2018). Here, we present *parent-scaled self-attention* (PASCAL): a novel, parameter-free local attention mechanism that lets the model focus on the dependency parent of each token when encoding the source sentence. Our method is simple yet effective, improving translation quality with no additional parameter or computational overhead.

Our main contributions are:

- introducing PASCAL: an effective parameter-free local self-attention mechanism to incorporate source-side syntax into Transformers;
- adapting LISA (Strubell et al., 2018) to sub-word representations and applying it to NMT;

[*]Work done while at Tokyo Institute of Technology.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1618–1627

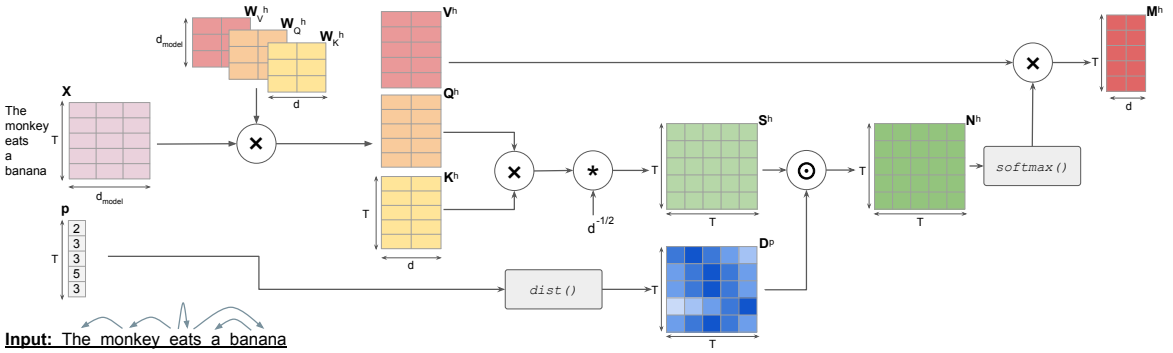

Figure 1: Parent-Scaled Self-Attention (PASCAL) head for the input sequence "The monkey eats a banana".

- similar to concurrent work (Pham et al., 2019), we find that modeling linguistic knowledge into the self-attention mechanism leads to better translations than other approaches.

Our extensive experiments on standard En↔De, En→Tr and En→Ja translation tasks also show that (a) approaches to embed syntax in RNNs do not always transfer to the Transformer, and (b) PASCAL consistently exhibits significant improvements in translation quality, especially for long sentences.

2 Model

In order to design a neural network that is efficient to train and that exploits syntactic information while producing high-quality translations, we base our model on the Transformer architecture (Vaswani et al., 2017) and upgrade its encoder with *parent-scaled self-attention* (PASCAL) heads at layer l_s. PASCAL heads enforce contextualization from the syntactic dependencies of each source token, and, in practice, we replace standard self-attention heads with PASCAL ones in the first layer as its inputs are word embeddings that lack any contextual information. Our PASCAL sub-layer has the same number H of attention heads as other layers.

Source syntax Similar to previous work, instead of just providing sequences of tokens, we supply the encoder with dependency relations given by an external parser. Our approach explicitly exploits sub-word units, which enable open-vocabulary translation: after generating sub-word units, we compute the middle position of each word in terms of number of tokens. For instance, if a word in position 4 is split into three tokens, now in positions 6, 7 and 8, its middle position is 7. We then map each sub-word of a given word to the middle position of its parent. For the root word, we define its parent to be itself, resulting in a parse that is a directed graph. The input to our encoder is a sequence of T tokens and the absolute positions of their parents.

2.1 Parent-Scaled Self-Attention

Figure 1 shows our parent-scaled self-attention sub-layer. Here, for a sequence of length T, the input to each head is a matrix $\mathbf{X} \in \mathbb{R}^{T \times d_{model}}$ of token embeddings and a vector $\mathbf{p} \in \mathbb{R}^T$ whose t-th entry p_t is the middle position of the t-th token's dependency parent. Following Vaswani et al. (2017), in each attention head h, we compute three vectors (called query, key and value) for each token, resulting in the three matrices $\mathbf{K}^h \in \mathbb{R}^{T \times d}$, $\mathbf{Q}^h \in \mathbb{R}^{T \times d}$, and $\mathbf{V}^h \in \mathbb{R}^{T \times d}$ for the whole sequence, where $d = d_{model}/H$. We then compute dot products between each query and all the keys, giving scores of how much focus to place on other parts of the input when encoding a token at a given position. The scores are divided by $\sqrt{d}$ to alleviate the vanishing gradient problem arising if dot products are large:

$$\mathbf{S}^h = \mathbf{Q}^h \, {\mathbf{K}^h}^\top / \sqrt{d}. \quad (1)$$

Our main contribution is in weighing the scores of the token at position t, $\mathbf{s}_t$, by the distance of each token from the position of t's dependency parent:

$$n^h_{tj} = s^h_{tj} \, d^p_{tj}, \quad \text{for } j = 1, ..., T, \quad (2)$$

where $\mathbf{n}^h_t$ is the t-th row of the matrix $\mathbf{N}^h \in \mathbb{R}^{T \times T}$ representing scores normalized by the proximity to t's parent. $d^p_{tj} = dist(p_t, j)$ is the $(t, j)^{th}$ entry of the matrix $\mathbf{D}^p \in \mathbb{R}^{T \times T}$ containing, for each row $\mathbf{d}_t$, the distances of every token j from the middle position of token t's dependency parent p_t. In this paper, we compute this distance as the value of the probability density of a normal distribution centered at p_t and with variance σ^2, $\mathcal{N}\left(p_t, \sigma^2\right)$:

$$dist(p_t, j) = f_{\mathcal{N}}\left(j | p_t, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(j-p_t)^2}{2\sigma^2}}. \quad (3)$$

Finally, we apply a softmax function to yield a distribution of weights for each token over all the tokens in the sentence, and multiply the resulting matrix with the value matrix $\mathbf{V}^h$, obtaining the final representations $\mathbf{M}^h$ for PASCAL head h.

One of the major strengths of our proposal is being parameter-free: no additional parameter is required to train our PASCAL sub-layer as $\mathbf{D}^p$ is obtained by computing a distance function that only depends on the vector of tokens' parent positions and can be evaluated using fast matrix operations.

Parent ignoring Due to the lack of parallel corpora with gold-standard parses, we rely on noisy annotations from an external parser. However, the performance of syntactic parsers drops abruptly when evaluated on out-of-domain data (Dredze et al., 2007). To prevent our model from overfitting to noisy dependencies, we introduce a regularization technique for the PASCAL sub-layer: *parent ignoring*. In a similar vein as dropout (Srivastava et al., 2014), we disregard information during the training phase. Here, we ignore the position of the parent of a given token by randomly setting each row of $\mathbf{D}^p$ to $\mathbf{1} \in \mathbb{R}^T$ with some probability q.

Gaussian weighing function The choice of weighing each score by a Gaussian probability density is motivated by two of its properties. First, its bell-shaped curve: It allows us to focus most of the probability density at the mean of the distribution, which we set to the middle position of the sub-word units of the dependency parent of each token. In our experiments, we find that most words in the vocabularies are not split into sub-words, hence allowing PASCAL to mostly focus on the actual parent. In addition, non-negligible weights are placed on the neighbors of the parent token, allowing the attention mechanism to also attend to them. This could be useful, for instance, to learn idiomatic expressions such as prepositional verbs in English. The second property of Gaussian-like distributions that we exploit is their support: While most of the weight is placed in a small window of tokens around the mean of the distribution, all the values in the sequence are actually multiplied by non-zero factors; allowing a token j farther away from the parent of token t, p_t, to still play a role in the representation of t if its score s^h_{tj} is high.

PASCAL can be seen as an extension of the local attention mechanism of Luong et al. (2015), with the alignment now guided by syntactic information. Yang et al. (2018) proposed a method to learn a Gaussian bias that is added to, instead of multiplied by, the original attention distribution. As we will see next, our model significantly outperforms this.

3 Experiments

3.1 Experimental Setup

Data We evaluate the efficacy of our approach on standard, large-scale benchmarks and on low-resource scenarios, where the Transformer was shown to induce poorer syntax. Following Bastings et al. (2017), we use News Commentary v11 (NC11) with En-De and De-En tasks to simulate low resources and test multiple source languages. To compare with previous work, we train our models on WMT16 En-De and WAT En-Ja tasks, removing sentences in incorrect languages from WMT16 data sets. For a thorough comparison with concurrent work, we also evaluate on the large-scale WMT17 En-De and low-resource WMT18 En-Tr tasks. We rely on Stanford CoreNLP (Manning et al., 2014) to parse source sentences.[1]

Training We implement our models in PyTorch on top of the Fairseq toolkit.[2] Hyperparameters, including the number of PASCAL heads, that achieved the highest validation BLEU (Papineni et al., 2002) score were selected via a small grid search.

We report previous results in syntax-aware NMT for completeness, and train a Transformer model as a strong, standard baseline. We also investigate the following syntax-aware Transformer approaches:[1]

- **+PASCAL:** The model presented in §2. The variance of the normal distribution was set to 1 (i.e., an effective window size of 3) as 99.99% of the source words in our training sets are at most split into 7 sub-words units.
- **+LISA:** We adapt LISA (Strubell et al., 2018) to NMT and sub-word units by defining the parent of a given token as its first sub-word (which represents the root of the parent word).
- **+MULTI-TASK:** Our implementation of the multi-task approach by Currey and Heafield (2019) where a standard Transformer learns to both parse and translate source sentences.
- **+S&H:** Following Sennrich and Haddow (2016), we introduce syntactic information in the form of dependency labels in the embedding matrix of the Transformer encoder.

[1] For a detailed description, see Appendix A.

[2] https://github.com/e-bug/pascal.

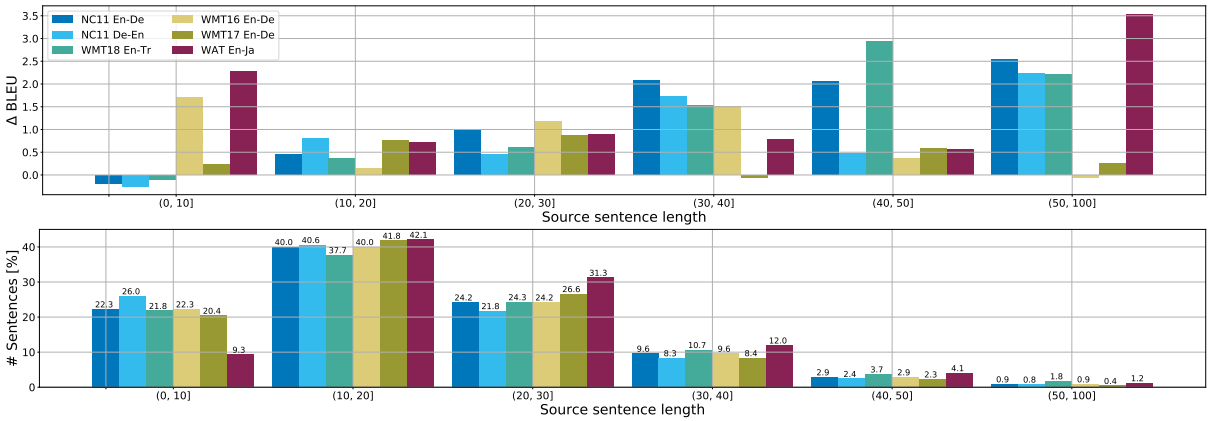

Figure 2: Analysis by sentence length: ΔBLEU with the Transformer (above) and percentage of data (below).

Method	NC11 En-De	NC11 De-En	WMT18 En-Tr	WMT16 En-De	WMT17 En-De	WAT En-Ja [B]	WAT En-Ja [R]
Eriguchi et al. (2016)						34.9	81.58
Bastings et al. (2017)	16.1						
Hashimoto and Tsuruoka (2017)						39.4	82.83
Bisk and Tran (2018)				30.3	24.3		
SE+SD-NMT† (Wu et al., 2018)					24.7	36.4	81.83
SE+SD-Transformer† (Wu et al., 2018)					**26.2**		
Mixed Enc. (Currey and Heafield, 2019)			9.6	31.9	26.0		
Multi-Task (Currey and Heafield, 2019)			10.6	29.6	23.4		
Transformer	25.0	26.6	13.1	33.0	25.5	43.1	83.46
+ PASCAL	**25.9**⇑	**27.4**⇑	**14.0**⇑	**33.9**⇑	*26.1*⇑	**44.0**⇑	**85.21**⇑
+ LISA	25.3	27.1	13.6	33.6	25.7	43.2	83.51
+ MULTI-TASK	24.8	26.7	**14.0**	32.4	24.6	42.7	84.18
+ S&H	25.5	26.8	13.0	31.9	25.1	42.8	83.88

Table 1: Test BLEU (and RIBES for En-Ja) scores on small-scale (left) and large-scale (right) data sets. Models that also require target-side syntax information are marked with †, while ⇑ indicates statistical significance ($p < 0.01$) against the Transformer baseline via bootstrap re-sampling (Koehn, 2004).

3.2 Results

Table 1 presents the main results of our experiments. Clearly, the base Transformer outperforms previous syntax-aware RNN-based approaches, proving it to be a strong baseline in our experiments. The table shows that the simple approach of Sennrich and Haddow (2016) does not lead to notable advantages when applied to the embeddings of the Transformer model. We also see that the multi-task approach benefits from better parameterization, but it only attains comparable performance with the baseline on most tasks. On the other hand, LISA, which embeds syntax in a self-attention head, leads to modest but consistent gains across all tasks, proving that it is also useful for NMT. Finally, PASCAL outperforms all other methods, with consistent gains over the Transformer baseline independently of the source language and corpus size: It gains up to +0.9 BLEU points on most tasks and a substantial +1.75 in RIBES (Isozaki et al., 2010), a metric with stronger correlation with human judgments than BLEU in En↔Ja translations. On WMT17, our slim model compares favorably to other methods, achieving the highest BLEU score across all source-side syntax-aware approaches.[3]

Overall, our model achieves substantial gains given the grammatically rigorous structure of English and German. Not only do we expect performance gains to further increase on less rigorous sources and with better parses (Zhang et al., 2019), but also higher robustness to noisier syntax trees obtained from back-translated with parent ignoring.

Performance by sentence length As shown in Figure 2, our model is particularly useful when translating long sentences, obtaining more than +2 BLEU points when translating long sentences in all low-resource experiments, and +3.5 BLEU points on the distant En-Ja pair. However, only a few sentences (1%) in the evaluation datasets are long.

[3]Note that modest improvements in this task should not be surprising as Transformers learn better syntactic relationships from larger data sets (Raganato and Tiedemann, 2018).

SRC	In a cooling experiment , **only** a tendency agreed
BASE	冷却 実験 で は , **わずか な** 傾向 が 一致 し た
OURS	冷却 実験 で は 傾向 **のみ** 一致 し た
SRC	Of course I **don't** hate you
BASE	Natürlich hass**te** ich dich nicht
OURS	Natürlich hass**e** ich dich nicht
SRC	What are those people fighting for?
BASE	Was sind die Menschen, für die kämpfen?
OURS	Wofür kämpfen diese Menschen?

Table 2: Example of correct translation by PASCAL.

Layer	En-De	De-En
1	**23.2**	**24.6**
2	22.5	20.1
3	22.5	23.8
4	22.6	23.8
5	22.9	23.8
6	22.4	23.9

(a)

Variance	En-De	De-En
Parent-only	22.5	22.4
1	**23.2**	**24.6**
4	22.7	24.3
9	22.8	24.3
16	22.7	24.4
25	22.8	24.1

(b)

Table 3: Validation BLEU as a function of PASCAL layer (a) and Gaussian's variance (b) on NC11 data.

Qualitative performance Table 2 presents examples where our model correctly translated the source sentence while the Transformer baseline made a syntactic error. For instance, in the first example, the Transformer misinterprets the adverb "only" as an adjective of "tendency:" the word "only" is an adverb modifying the verb "agreed." In the second example, "don't" is incorrectly translated to the past tense instead of present.

PASCAL layer When we introduced our model, we motivated our design choice of placing PASCAL heads in the first layer in order to enrich the representations of words from their isolated embeddings by introducing contextualization from their parents. We ran an ablation study on the NC11 data in order to verify our hypothesis. As shown in Table 3a, the performance of our model on the validation sets is lower when placing Pascal heads in upper layers; a trend that we also observed with the LISA mechanism. These results corroborate the findings of Raganato and Tiedemann (2018) who noticed that, in the first layer, more attention heads solely focus on the word to be translated itself rather than its context. We can then deduce that enforcing syntactic dependencies in the first layer effectively leads to better word representations, which further enhance the translation accuracy of the Transformer model. Investigating the performance of multiple syntax-aware layers is left as future work.

Gaussian variance Another design choice we made was the variance of the Gaussian weighing function. We set it to 1 in our experiments motivated by the statistics of our datasets, where the vast majority of words is at most split into a few tokens after applying BPE. Table 3b corroborates our choice, showing higher BLEU scores on the NC11 validation sets when the variance equals 1. Here, "parent-only" is the case where weights are only placed to the middle token (i.e. the parent).

Sensitivity to hyperparameters Due to the large computational cost required to train Transformer models, we only searched hyperparameters in a small grid. In order to estimate the sensitivity of the proposed approach to hyperparameters, we trained the NC11 De-En model with the hyperparameters of the En-De one. In fact, despite being trained on the same data set, we find that more PASCAL heads help when German (which has a higher syntactic complexity than English) is used as the source language. In this test, we only find -0.2 BLEU points with respect to the score listed in Table 1, showing that our general approach is effective regardless of extensive fine-tuning.

Additional analyses are reported in Appendix B.

4 Conclusion

This study provides a thorough investigation of approaches to induce syntactic knowledge into self-attention networks. Through extensive evaluations on various translation tasks, we find that approaches effective for RNNs do not necessarily transfer to Transformers (e.g. +S&H). Conversely, dependency-aware self-attention mechanisms (LISA and PASCAL) best embed syntax, for all corpus sizes, with PASCAL consistently outperforming other all approaches. Our results show that exploiting core components of the Transformer to embed linguistic knowledge leads to higher and consistent gains than previous approaches.

Acknowledgments

We are grateful to the anonymous reviewers, Desmond Elliott and the CoAStaL NLP group for their constructive feedback. 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.

References

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. 2017. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967. Association for Computational Linguistics.

Yonatan Bisk and Ke Tran. 2018. Inducing Grammars with and for Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 25–35, Melbourne, Australia. Association for Computational Linguistics.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945. Association for Computational Linguistics.

Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2018. Syntax-directed attention for neural machine translation. In *AAAI Conference on Artificial Intelligence*.

Anna Currey and Kenneth Heafield. 2019. Incorporating source syntax into transformer-based neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 24–33, Florence, Italy. Association for Computational Linguistics.

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, João Graca, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1051–1055, Prague, Czech Republic. Association for Computational Linguistics.

Jeffrey L. Elman. 1990. Finding Structure in Time. *Cognitive Science*, 14(2):179–211.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-Sequence Attentional Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 823–833, Berlin, Germany. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2017. Neural Machine Translation with Source-Side Latent Graph Parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 125–135. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thuong Hai Pham, Dominik Macháček, and Ondřej Bojar. 2019. Promoting the knowledge of source syntax in transformer nmt is not needed. *Computación y Sistemas*, 23(3).

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

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.

Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2018. Multi-representation ensembles and delayed SGD updates improve syntax-based NMT. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 319–325, Melbourne, Australia. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic Input Features Improve Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Lei Shi, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A DOM tree alignment model for mining parallel data from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 489–496, Sydney, Australia. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. 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.

Ke Tran, Arianna Bisazza, and Christof Monz. 2018. The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4731–4736, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for Neural Machine Translation. *CoRR*, abs/1803.07416.

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 *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Shuangzhi Wu, Dongdong Zhang, Zhirui Zhang, Nan Yang, Mu Li, and Ming Zhou. 2018. Dependency-to-dependency neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 2132–2141.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Baosong Yang, Jian Li, Derek F. Wong, Lidia S. Chao, Xing Wang, and Zhaopeng Tu. 2019. Context-aware self-attention networks. In *AAAI Conference on Artificial Intelligence*.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458, Brussels, Belgium. Association for Computational Linguistics.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2019. Syntax-enhanced neural machine translation with syntax-aware word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1151–1161, Minneapolis, Minnesota. Association for Computational Linguistics.

Corpus		Train	Filtered Train	Valid	Test
NC11	En-De	238,843	233,483	2,169	2,999
WMT18	En-Tr	207,373		3,000	3,007
WMT16	En-De	4,500,962	4,281,379	2,169	2,999
WMT17	En-De	5,852,458		2,999	3,004
WAT	En-Ja	3,008,500		1,790	1,812

Table 4: Number of sentences in each data set.

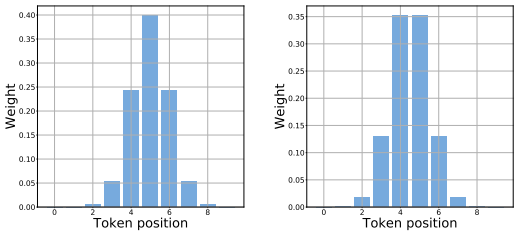

Figure 3: Weights of normal probability density with $\sigma^2 = 1$ and the means at positions 5 (left) or 4.5 (right).

A Experiment details

Data preparation We follow the same preprocessing steps as Vaswani et al. (2017). Unless otherwise specified, we first tokenize the data with Moses (Koehn et al., 2007) and remove sentences longer than 80 tokens in either source or target side.

Following Bastings et al. (2017), we train on News Commentary v11 (NC11) data set with English→German (En-De) and German→English (De-En) tasks so as to simulate low-resource cases and to evaluate the performance of our models for different source languages. We also train on the full WMT16 data set for En-De, using *newstest2015* and *newstest2016* as validation and test sets, respectively, in each of these experiments. Moreover, we notice that these data sets contain sentences in different languages and use `langdetect`[4] to remove sentences in incorrect languages.

We also train our models on WMT18 English→Turkish (En-Tr) as a standard low-resource scenario. Models are evaluated on *newstest2016* and tested on *newstest2017*.

Previous studies on syntax-aware NMT have commonly been conducted on the WMT16 En-De and WAT English→Japanese (En-Ja) tasks, while concurrent approaches are evaluated on the WMT17 En-De task. In order to provide a generic and comprehensive evaluation of our proposed approach on large-scale data, we also train our models on the latter tasks. We follow the WAT18 preprocessing steps[5] for experiments on En-Ja but use Cabocha[6] to tokenize target sentences. On WMT17, we use *newstest2016* and *newstest2017* as validation and test sets, respectively.

Table 4 lists the final sizes of each data set.

Baselines We evaluate the impact of syntactic information with the following approaches:

- **Transformer:** We train a base Transformer model as a strong, standard baseline using the hyperparameters in the latest Tensor2Tensor (Vaswani et al., 2018) version (3).
- **+S&H:** Following Sennrich and Haddow (2016), we introduce syntactic information in the form of dependency labels in the embedding matrix of the Transformer encoder. More specifically, each token is associated with its dependency label which is first embedded into a vector representation of size 10 and then used to replace the last 10 embedding dimensions of the token embedding, ensuring a final size that matches the original one.
- **+MULTI-TASK:** Our implementation of the multi-task approach by Currey and Heafield (2019) where a standard Transformer learns to both parse and translate source sentences. Each source sentence is first duplicated and associated its linearized parse as target sequence. To distinguish between the two tasks, a special tag indicating the desired task is prepended and appended to each source sentence. Finally, parsing and translation training data is shuffled together.
- **+LISA:** We adapt Linguistically-Informed Self-Attention (LISA; Strubell et al. 2018) to NMT. In one attention head h, $\mathbf{Q}^h$ and $\mathbf{K}^h$ are computed through a feed-forward layer and the key-query dot product to obtain attention weights is replaced by a bi-affine operator $\mathbf{U}$. These attention weights are further supervised to attend to each token's parent by interpreting each row t as the distribution over possible parents for token t. Here, we extend the authors' approach to BPE by defining the parent of a given token as its first sub-word unit (which represents the root of the parent word). The model is trained to maximize the joint probability of translations and parent positions.

[4] https://pypi.org/project/langdetect.
[5] http://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2018/baseline/dataPreparationJE.html.
[6] https://taku910.github.io/cabocha/.

Component	NC11 En-De	NC11 De-En	WMT18 En-Tr	WMT16 En-De	WMT17 En-De	WAT En-Ja
Transformer	22.6	23.8	12.6	29.0	31.5	42.2
+ data filtering	22.8 (+0.2)	24.0 (+0.2)		28.7 (-0.3)		
+ PASCAL	23.0 (+0.2)	24.6 (+0.6)	13.6 (+1.0)	29.2 (+0.5)	31.6 (+0.1)	43.5 (+1.3)
+ parent ignoring	23.2 (+0.2)		13.7 (+0.1)		32.1 (+0.6)	

Table 5: Validation BLEU when incrementally adding each component used by our best-performing models.

Corpus	Transformer	+PASCAL
NC11 En-De	4,134.1	4,188.8
NC11 De-En	4,276.6	4,177.4
WMT18 En-Tr	3,559.7	3,621.1
WMT16 En-De	23,186.3	23,358.8
WMT17 En-De	23,604.1	24,083.6
WAT En-Ja	23,005.8	23,073.0

Table 6: Training times (in seconds) for the Transformer baseline and Transformer+PASCAL on each data set. PASCAL adds negligible overhead.

Corpus	lr	(β_1, β_2)	h_P	q
NC11 En-De	0.0007	(0.9, 0.997)	2	0.4
NC11 De-En	0.0007	(0.9, 0.997)	8	0.0
WMT18 En-Tr	0.0007	(0.9, 0.980)	7	0.3
WMT16 En-De	0.0007	(0.9, 0.980)	5	0.0
WMT17 En-De	0.0007	(0.9, 0.997)	7	0.3
WAT En-Ja	0.0007	(0.9, 0.997)	7	0.0

Table 7: Hyperparameters for the reported models. lr denotes the maximum learning rate, (β_1, β_2) are Adam's decay rates, h_P is the number of PASCAL heads, and q is the parent ignoring probability.

Training details All experiments are based on the base Transformer architecture and optimized following the learning schedule of Vaswani et al. (2017) with $8,000$ warm-up steps. Similarly, we use label smoothing $\epsilon_{ls} = 0.1$ (Szegedy et al., 2016) during training and employ beam search with a beam size of 4 and length penalty $\alpha = 0.6$ (Wu et al., 2016) at inference time. We use a batch size of $32K$ tokens and run experiments on a cluster of 4 machines, each having 4 Nvidia P100 GPUs. See Table 6 for the training times of each experiment.

For each model, we run a small grid search over the hyperparameters and select the ones giving the highest BLEU scores on validation sets (Table 7).

We use the SACREBLEU (Post, 2018) tool to compute case-sensitive BLEU scores.[7] When evaluating En-Ja translations, we follow the procedure employed at WAT by computing BLEU scores after tokenizing target sentences using KyTea.[8]

[7]Signature: BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.2.12.
[8]http://www.phontron.com/kytea/.

Following Vaswani et al. (2017), we train Transformer-based models for $100K$ steps on large-scale data. On small-scale data, we train for $20K$ steps and use a dropout probability $P_{drop} = 0.3$ as they let the Transformer baseline achieve higher performance on this size of data. For instance, in WMT18 En-Tr, our baseline outperforms the one in Currey and Heafield (2019) by $+3.5$ BLEU.

B Analysis

Multiplication vs. addition In Equation (2), we calculated the weighing scores by multiplying the self-attention scores by the distance to the parent token. Multiplication is, in fact, the standard way to weight values (e.g., the gating mechanism of LSTMs and GRUs). In our case, it introduces sparseness in the attention scores for non-parent tokens. Moreover, it weights gradients in back-propagation: Let x and y be the attention score and dependency weight, respectively. Consider a loss $l = f(z)$ where $z = xy$ and $dl/dx = df(z)/dz * y$. The attention score receive gradients more on dependent pairs (larger y) than non-dependent ones (smaller y), which is sound for dependency information. In contrast, addition cannot obtain such an effect because it does not affect gradients: $dl/dx = df(z)/dz$ when $z = x + y$. For completeness, we trained our best NC11 models replacing multiplication by addition. We find that BLEU scores still improve upon the baseline, meaning that our approach is robust, but find them to be slightly lower (-0.2) than with multiplication.

Ablation We introduced different techniques to improve neural machine translation with syntax information. Table 5 lists the contribution of each technique, in an incremental fashion, whenever they were used by the models reported in Table 1.

While removing sentences whose languages do not match the translation task can lead to better performance (NC11), the precision of the detection tool assumes a major role at large scale. In WMT16, `langdetect` removes more than $200K$ sentences and leads to performance losses. It would

also drop $19K$ pairs on the clean WAT En-Ja data.

The proposed PASCAL mechanism is the component that most improves the performance of the models, achieving up to $+1.0$ and $+1.3$ BLEU on the distant En-Tr and En-Ja pairs, respectively. With the exception of NC11 En-De, we find parent ignoring useful on the noisier WMT18 En-Tr and WMT17 En-De datasets. In the former, low-resource case, the benefits of parent ignoring are minimal, but it proves fundamental on the large-scale WMT17 data, where it leads to significant gains when paired with the PASCAL mechanism.[9]

Finally, looking at the number of PASCAL heads in Table 7, we notice that most models rely on a large number of syntax-aware heads. Raganato and Tiedemann (2018) found that only a few attention heads per layer encoded a significant amount of syntactic dependencies. Our study shows that the Transformer model can be improved by having more attention heads learn syntactic dependencies.

[9]Note that this ablation is obtained by stripping away each component from the best performing models and hence only seeing $+0.1$ for PASCAL on WMT17 En-De does not mean that PASCAL is not helpful in this task but rather that combining it with parent ignoring gives better performance (our second best model achieved $+0.5$ by using PASCAL only).