NLP Applications Project

Incorporating Dependency Syntax into Transformer-based NMT

Ujwal Narayan
20171170

Saujas Vaduguru
20171098

Abstract

Transformer-based models have surpassed RNN-based neural machine translation models in terms of performance. While they achieve high scores when trained on large amounts of training data, there is still scope for improvement in their use in low- and moderate-resource settings. We propose incorporating syntax information explicitly into the training process using dependency parse trees. We explore two methods of augmenting the training data with parse trees, and report results on subsets of the Europarl corpus. We also evaluate the self-attention matrices for syntactic representation by inducing dependency trees from them. We find that in this training regime, the use of explicit syntactic information does not improve the performance of the Transformer models in a low-resource setting. Our code is available at `https://github.com/saujasv/dependent-transformers`.[1]

1 Introduction

Recent work in neural machine translation has been dominated by architectures based on the Transformer model (Vaswani et al., 2017). These models achieved state-of-the-art performance on the WMT 2014 English-to-German and English-to-French tasks, and outperformed RNN-based approaches to neural machine translation.

While Transformers perform well when trained on large corpora (WMT 2014 English-to-German has 4.5M sentence pairs, and English-to-French has 36M sentence pairs), the application of Transofrmers to low-resource and moderate-resource settings remains an area of active exploration.

One direction for improving performance in lower resource settings that has been explored is providing transformers with explicit syntactic information. Previous work by Tran et al. (2018) comapring the representations learned by Transformer models to those by recurrent models has shown that recurrent models are better at capturing hierarchical information. This presents a stronger case for using explicit syntactic information to enhance Transformer models.

Currey and Heafield (2019) are among the first to explore this line of work, using constituency parse information to augment the training data provided to Transformer models. They propose two mechanisms to supplement training data with syntactic information, which we adopt here. They find that while the additional syntactic information prdocues mixed results in a high-resource setting, it allows models to consistently outperform a non-syntactic baseline in low-resource settings.

Instead of constituency parse information, we explore the possibility of using dependency parse information in the training data. We explore both forms of augmentation proposed by Currey and Heafield (2019). To analyse the results of adding syntactic information to our training regime, we use one of the techniques proposed by Raganato and Tiedemann (2018). We use the attention heads in the encoder of the Transformer model to induce dependency trees, and evaluate results on a dependency parsing benchmark.

[1] Our data and trained models are available at `http://tiny.cc/dep-transformer-files`

2 Related Work

There have been multiple approaches to incorporating syntax explicitly into Transfomer-based representations. One of them – by Currey and Heafield (2019) – is the inspiration for our work, and works with constituency parse trees.

There are also works that explore using dependency trees as syntactic input to the Transformer model. Omote et al. (2019) incorporate dependency information into the positional encoding using pairwise relative depths instead of pairwise relative positions. Strubell et al. (2018) introduce a modified self-attention mechanism – linguistically-informed self-attention (LISA) – that uses the dependency structure of the source explicitly in the calculation of self-attention. They train their models in a multi-task setting, but do not include machine translation as one of the tasks. Deguchi et al. (2019) extend LISA to machine translation. Zhang et al. (2019) use hidden representations from a neural dependency parser as the input to the encoder of an NMT model.

There has also been work on incorporating linearized parses as explicit syntactic information in RNN-based NMT models. Wu et al. (2017) use linearized dependency trees in the same manner we do, but also explore multiple ways of traversing the tree to linearize it. We use one of the schemes they propose – the Child Enriched Structure. Aharoni and Goldberg (2017) and Currey and Heafield (2018) propose incorporating linearized constituency parses as explicit syntactic information in a machine translation system. These latter two approaches are also the ones that inform Currey and Heafield (2019).

3 Methods

3.1 Linearizing parses

We experiment with both constituency parse trees and dependency parse trees. We use the linearized constituency parses produced by the Stanford CoreNLP package. These are depth-first traversals of the parse tree, and we tokenize the phrase label associated with a phrase with the opening parentheses associated with the phrase.

We linearize dependency parses also as a depth-first tree traversal, and tokenize the opening parenthesis of each subtree with the dependency label that joins the governor to that subtree.

Sentence	Parse Type	Parse
It is only natural!	Dependency Parse	`(_ROOT natural (_nsubj It ) (_cop is ) (_advmod only ) (_punct ! ) ) )`
	Constituency Parse	`(ROOT (S (NP (PRP It)) (VP (VBZ is) (ADJP (RB only) (JJ natural))) (. !)))`

Table 1: Examples of linearized parses for constituency and dependency parses.

3.2 Multi-task

The first method of presenting additional syntactic information is to train the model as a multi-task model, where one task is to translate the sentence, and the other is to parse it. However, unlike some multi-task approaches (Subramanian et al., 2018) to improving source-side representations in encoder-decoder architectures, we do not use a different decoder. The same encoder and decoder modules are trained to perform both tasks. Each sentence is presented as `<TAG> Sentence <TAG>`, where `<TAG>` specifies the task. We expect that the encoder learns to better represent the hierarchical structure of a sentence during encoding, and the decoder learns to exploit the hierarchical structure of the sentence during translation. During training, the model is presented with the same sentence twice – once with the task of translation, and once with the task of parsing.

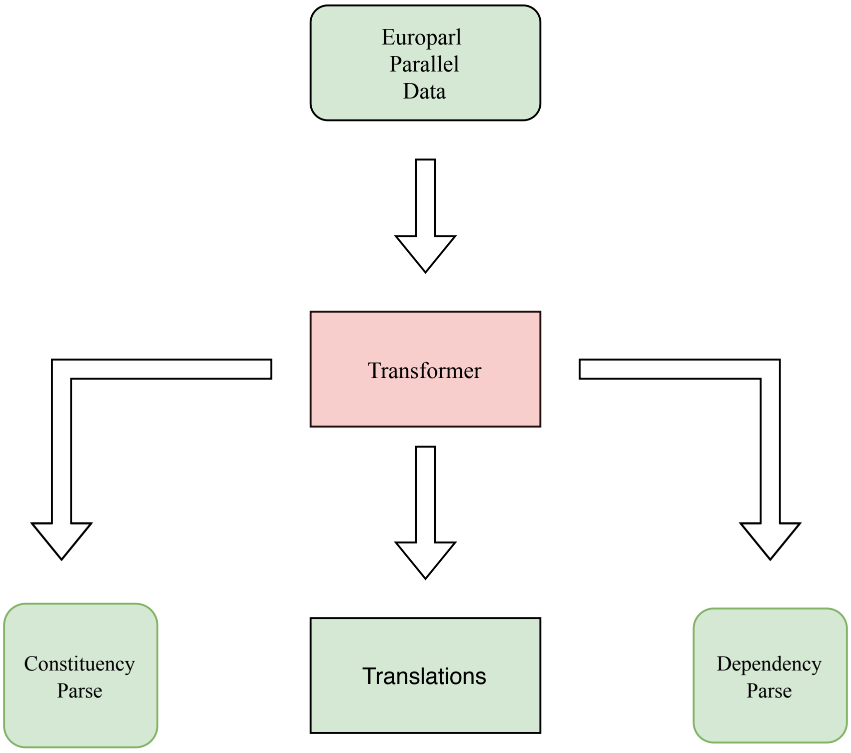


Figure 1: The multi-task training pipeline

3.3 Mixed encoder

The second method we use to introduce syntactic information is to present only one kind of target data (as opposed to the multi-task setting, where the target may be the translation, or a linearized parse) and vary only the source sequence. In the mixed encoder setting, the model is presented with either a sentence, or its linearized parse tree, and tasked with determining the translation. During training, the model is presented with the same sentence twice – once as the sentence itslef, and once as the parse tree – and the target sequence in both cases is the translation. In this setting, we expect the encoder to learn hierarchical representations better since it is presented with both linear and hierarchical information.

3.4 Dependency tree induction

To examine the changes brought about by introducing more syntactic information being provided, we examine the attention heads for representation of syntax. To do this, we use the method adopted by Raganato and Tiedemann (2018), which is using the attention heads to induce dependency trees.

We can view attention heads as matrices that represent the adjecency matrix of a complete graph in which each word in the sentence is a vertex. The values in the matrix represent the strength of the connection. Raganato and Tiedemann (2018) propose constructing dependency trees as maximum spanning trees of the graph. The Chu-Liu-Edmonds (i Chu and Liu, 1965; Edmonds, 1967) algorithm is used to obtain the spanning tree.

While far from sufficient for parsing, this allows us to examine the syntactic relations that the attention matrices capture, and the impact of syntactic information on the attention matrices.

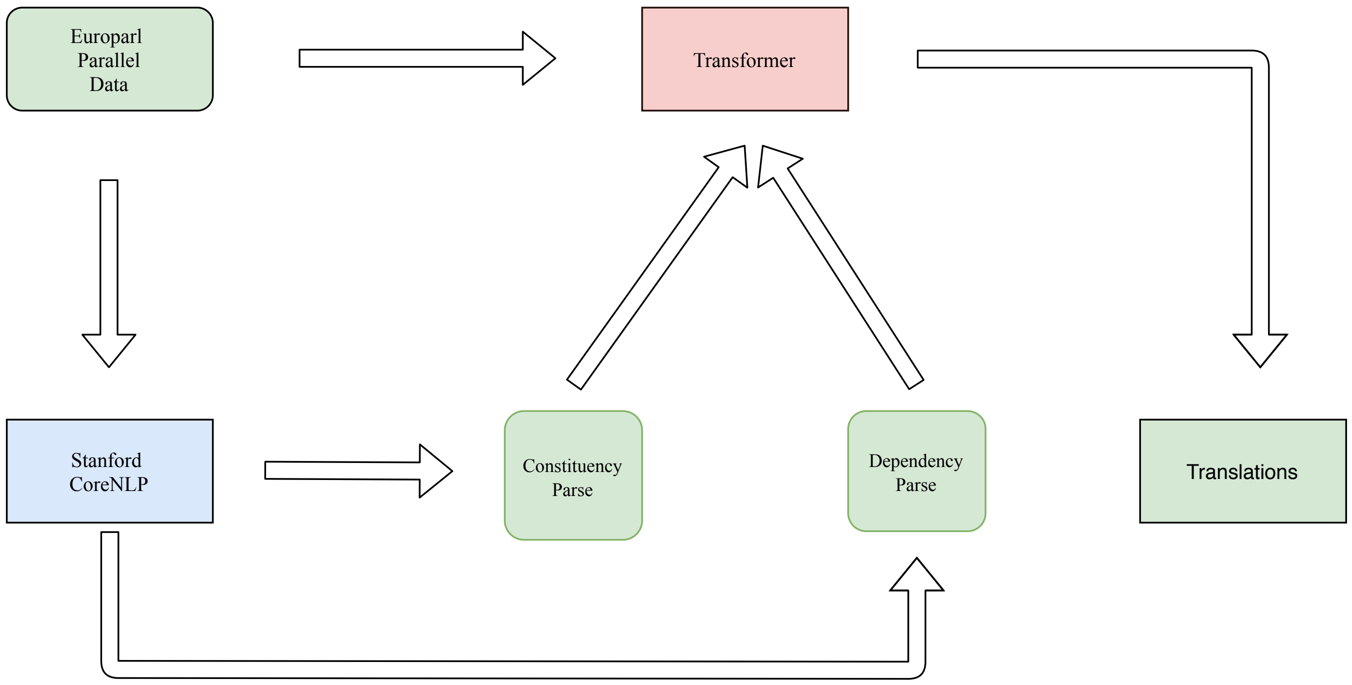


Figure 2: The mixed encoder training pipeline

4 Experiments

4.1 Data

We carried out experiments for the following language pairs

- English - Finnish
- English - German

The data for these were sourced from the Europarl corpus (Koehn, 2005). These languages are strongly agglutenative and we believe that these languages will benifit the most from introducing the dependency relations. We also chose representatives of different language families, Finnish being Uralic, and German being Indo-European.

From the corpus, we collected the sentences at the intersection of all the language pairs. Repeated sentneces were purned from the dataset to prevent data contamination i.e the leaking of the train data samples to test and valid sets. The sentences were then parsed with Stanford CoreNLP parser and the resulting dependency trees and constituency parse tree were extracted. Sentences with embedded quotes (around 15k in number) were removed from the dataset resulting in **467737** sentence pairs for each language. The data was then split it into train, development, and test in roughly an *80:10:10* ratio resulting in the following numbers.

Split	Number of Sentences
Train	374112
Valid	46772
Test	46853

Table 2: Dataset split

For the multi task setting, tokens were appended to both the beginning and the end of the sentences to indicate the target tasks i.e what kind of output was to be generated.

Token	Target Task
<TR>	Translate the sentence
<CP>	Generate the constituency parse tree for the sentence
<DP>	Generate the dependency parse tree for the sentence

Table 3: Target task in indicators in the multi-task setting

4.2 Model

We used Klein et al. (2017)'s OpenNMT implementation of the transformer to train all our models. We stuck to the same hyperparamters as that of Vaswani et al. (2017). As stated earlier there are two major types of experiments.

- Multi Task
- Mixed Encoder

For each of these two types we carreied out experiments for all combinations of presence and abscense of our features namely constituency parse trees and dependency parse trees, resulting in three models per experiment.

For generating the translations we used the same Klein et al. (2017) OpenNMT framework, and stuck to the same hyperparameters as that of Vaswani et al. (2017). A slight modification was done to the testing corpus on the source side for multi task encoders. The token `<TR>` was appended to the start and end of all sentences to make the model predict the translations instead of the parse trees.

5 Results and Discussion

We evalauted the translations using Post (2018)'s implementation of Papineni et al. (2002)'s `BLEU` metric. From tables 4a and 4b, we can see that the base models without any source syntax injections outperform all the other models by a huge margin. The addition of syntax seems to have catostiphically affected the translation quality.

Qualitatively observing the data, we see that the same kind of sentences seems to be repeated multiple times. Phrases such as

> *Ich habe für diesen Bericht gestimmt, da ich der Ansicht bin, dass die Europäische Union eine.*
>
> Translation: I voted for this report because I believe that the European Union is one.

occurs very often in the predictions. The Finnish side was frequently plauged by `<unk>` tokens causing poor BLEU scores.

We also observe that multi-task models perform much better in comparison to mixed-encoder models. And of course, as stated earlier they are all vastly inferior to the base models. The addition of syntax to the source side seems to have appeared as noise to the transformer, and the fact that multi-task models outperform mixed-encoder models certainly helps in making a case for this. There seems to be overfitting and a coverage issue with respect to the English-German pair whereas in the Finnish case, there seems to have the opposite effect with the predictions averaging two `<unk>` tokens per sentence.

5.1 Analysis

We further proceed to analyse the performance on two fronts.

Model Type	CP	DP	BLEU Score
Base	−	−	28.42
Multi Task	+	−	0.72
	+	+	0.61
	−	+	0.73
Mixed Encoder	+	-	0.67
	+	+	0.75
	-	+	0.43

(a) English-German

Model Type	CP	DP	BLEU Score
Base	−	−	18.61
Multi Task	+	−	18.34
	+	+	17.53
	−	+	17.71
Mixed Encoder	+	−	8.72
	+	+	8.11
	−	+	8.03

(b) English-Finnish

Table 4: BLEU scores different models

1. Translation performance against the complexity of the sentence measured using proxy indicators such as the length of the sentence, the depth of the constiuency parse trees and the depth of the dependency parse trees

2. We also use dependency tree induction as described in Section 3 as an analysis tool. We use the encoder attention heads to induce trees for the training set of the CoNLL 2017 Shared Task (Zeman et al., 2017), and compare unlabelled attachment scores (UAS) across layers and attention heads. We use the gold word and sentence tokenization.

5.1.1 Performance over complexity

We group together sentences of the same length , sentences with the same depth and evaluate their BLEU results. A bucket of size 1 was chosen to provide a holisitc view. One possible disadavantage with a fine bucket size is the lack of examples per bucket. But as we can see, there were sufficient examples for most of the buckets. Figure 3 shows the distribution of the sentences over the length of the sentences and the depth of their parse trees.

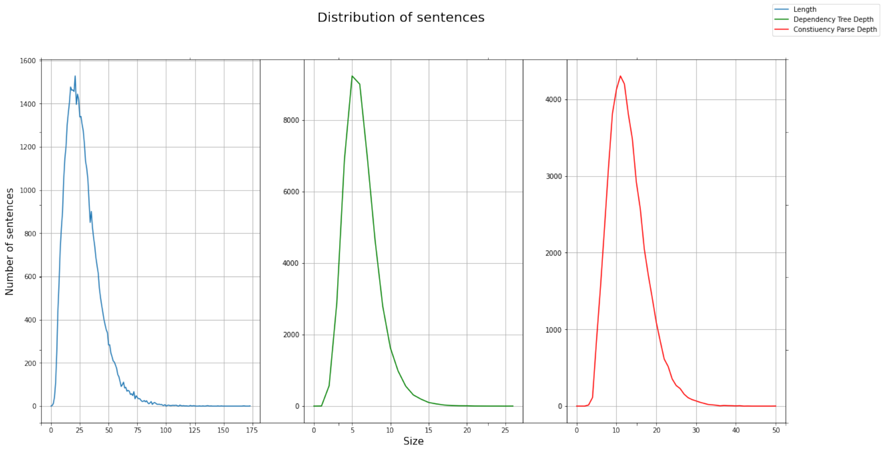


Figure 3: Sentence complexity of the test sentences

We observe that the degradation of performance with increasing length, and depth of dependency or constituency tree is similar in the case of base model and models we train on augmented data. We do not observe better performance on more complex (longer, or deeper) sentences when we provide explicit syntactic information. In interest of space, we leave the graphs showing variation in performance with length to the appendix (7).

5.1.2 Tree induction

For each model, we report the UAS F1 score achieved by the best performing attention head at each layer. We see that for German, using constituency parses leads to less powerful syntactic representations, while using only dependency parses brings it closer to the performance of the base model.

For Finnish, using both constituency and dependency parses works well in some layers, while using either one results in largely similar performance.

Note that these UAS scores are far lower than what can be acieved by a dedicated parsing model, and are presented here to help compare the syntactic information in the representations learned by the models. Raganato and Tiedemann (2018) note that when given more data, Transformers tend to learn better syntactic representations. We see that providing explicit syntactic information does not change this trend, and in a lower resource setting, Transformers do no capture syntax as well as they do when trained on large corpora.

Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6
Base	20.68	15.96	14.66	18.87	20.67	7.08
Base + CP	5.05	3.88	4.27	5.97	4.83	6.22
Base + CP + DP	14.64	5.08	5.22	11	7.66	7.07
Base + DP	15.63	14.84	13.13	5.06	5.59	15.57

(a) Best performing attention heads for English-German

Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6
Base	19.08	21.12	7.04	17.84	15.07	20
Base + CP	14.8	16.22	15.47	19.95	19.4	18.95
Base + CP + DP	17.73	13.67	20.31	8.62	16.07	14.5
Base + DP	14.14	17.94	14.9	19.97	17.47	7.37

(b) Best performing attention heads for English-Finnish

Table 5: UAS F1-scores on the dependency tree induction task on the CoNLL 2017 Shared Task English data.

6 Conclusion

In this project, we explore two methods of augmenting the training data provided to a Transformer machine translation model to improve the performance on smaller data sets. The first method involves training Transformers as multi-task models that predict the linearized parse trees and translations to the target language. The other method involves training the Transformer to generate the translation from either the source sentence or its parse tree. We propose the use of linearized dependency trees as the additional data for training the models.

We find that in this training regime, the use of explicit syntactic information does not improve the performance of the Transformer models in a low-resource setting. Even further analysis to induce dependency trees from attention heads shows that training with syntax data does not improve the encoder representations learned by the models.

7 Future Work

We propose two possible avenues for future work. The first is training the multi-task model similar to Subramanian et al. (2018). Our work explores a multi-task model that shares both the encoder and the decoder. However, using different decoders might allow the translation decoder to learn to generate translations better, without having to learn another language.

Another possible question to explore is the use of gold parses as the explicit syntactic information provided. The parses we use are generated by another model, and errors in parsing are likely to propogate and get amplified. However, incorporating gold parses might help mitigate this issue.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 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*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Ke Tran, Arianna Bisazza, and Christof Monz. 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, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1503. URL https://www.aclweb.org/anthology/D18-1503.

Anna Currey and Kenneth Heafield. 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, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5203. URL https://www.aclweb.org/anthology/W19-5203.

Alessandro Raganato and Jörg Tiedemann. 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, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5431. URL https://www.aclweb.org/anthology/W18-5431.

Yutaro Omote, Akihiro Tamura, and Takashi Ninomiya. Dependency-based relative positional encoding for transformer NMT. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 854–861, Varna, Bulgaria, September 2019. INCOMA Ltd. doi: 10.26615/978-954-452-056-4_099. URL https://www.aclweb.org/anthology/R19-1099.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1548. URL https://www.aclweb.org/anthology/D18-1548.

Hiroyuki Deguchi, Akihiro Tamura, and Takashi Ninomiya. Dependency-based self-attention for transformer NMT. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 239–246, Varna, Bulgaria, September 2019. INCOMA Ltd. doi: 10.26615/978-954-452-056-4_028. URL https://www.aclweb.org/anthology/R19-1028.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1118. URL https://www.aclweb.org/anthology/N19-1118.

Shuangzhi Wu, Ming Zhou, and Dongdong Zhang. Improved neural machine translation with source syntax. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4179–4185, 2017. doi: 10.24963/ijcai.2017/584. URL https://doi.org/10.24963/ijcai.2017/584.

Roee Aharoni and Yoav Goldberg. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2021. URL https://www.aclweb.org/anthology/P17-2021.

Anna Currey and Kenneth Heafield. Multi-source syntactic neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2961–2966, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1327. URL https://www.aclweb.org/anthology/D18-1327.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B18WgG-CZ.

Yau i Chu and T. Liu. On the shortest arborescence of a directed graph. 1965.

Jack Edmonds. Optimum branchings. 1967.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer, 2005.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*, 2017. doi: 10.18653/v1/P17-4012. URL https://doi.org/10.18653/v1/P17-4012.

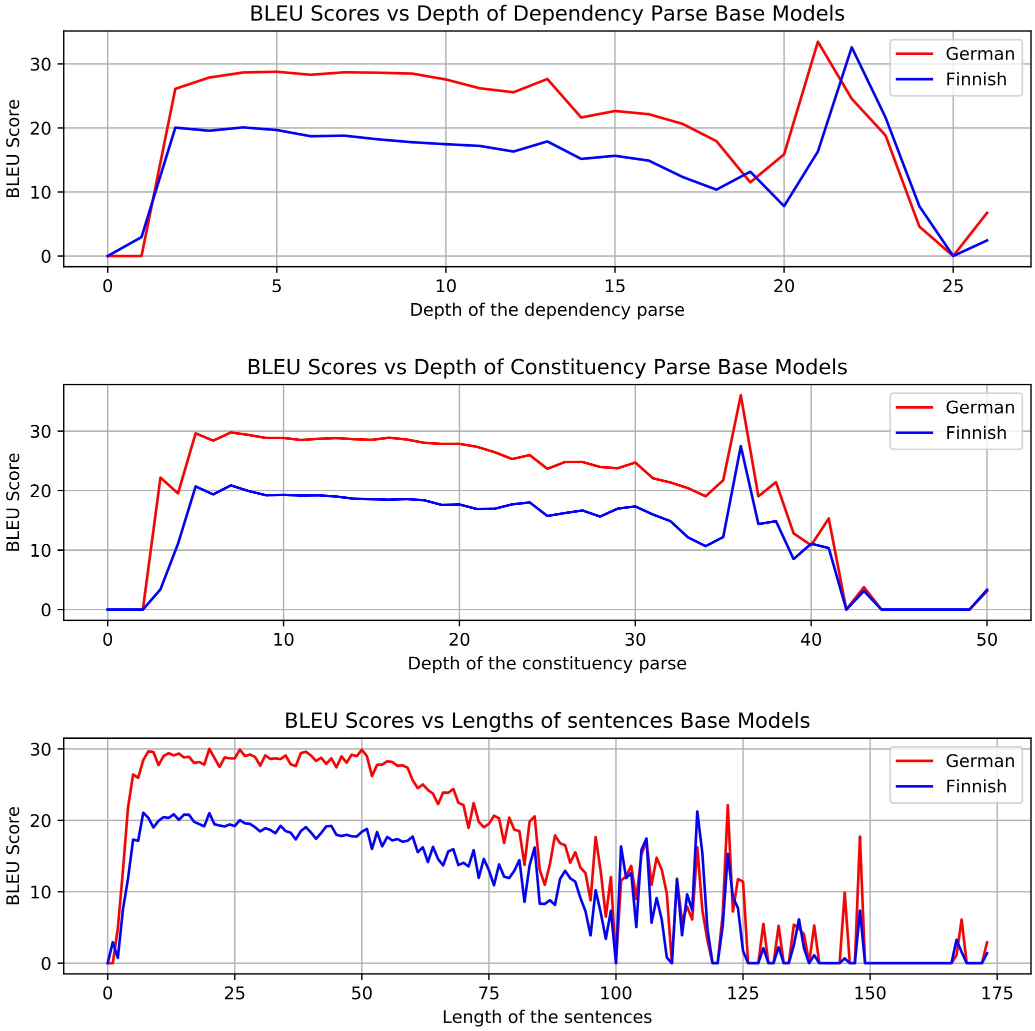
Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada, August 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/K/K17/K17-3001.pdf.

Appendix

Performance over Complexity Graphs



BLEU Scores vs Depth of Dependency Parse Base Models

BLEU Scores vs Depth of Constituency Parse Base Models

BLEU Scores vs Lengths of sentences Base Models

Figure 4: Base Models performance over sentence complexities

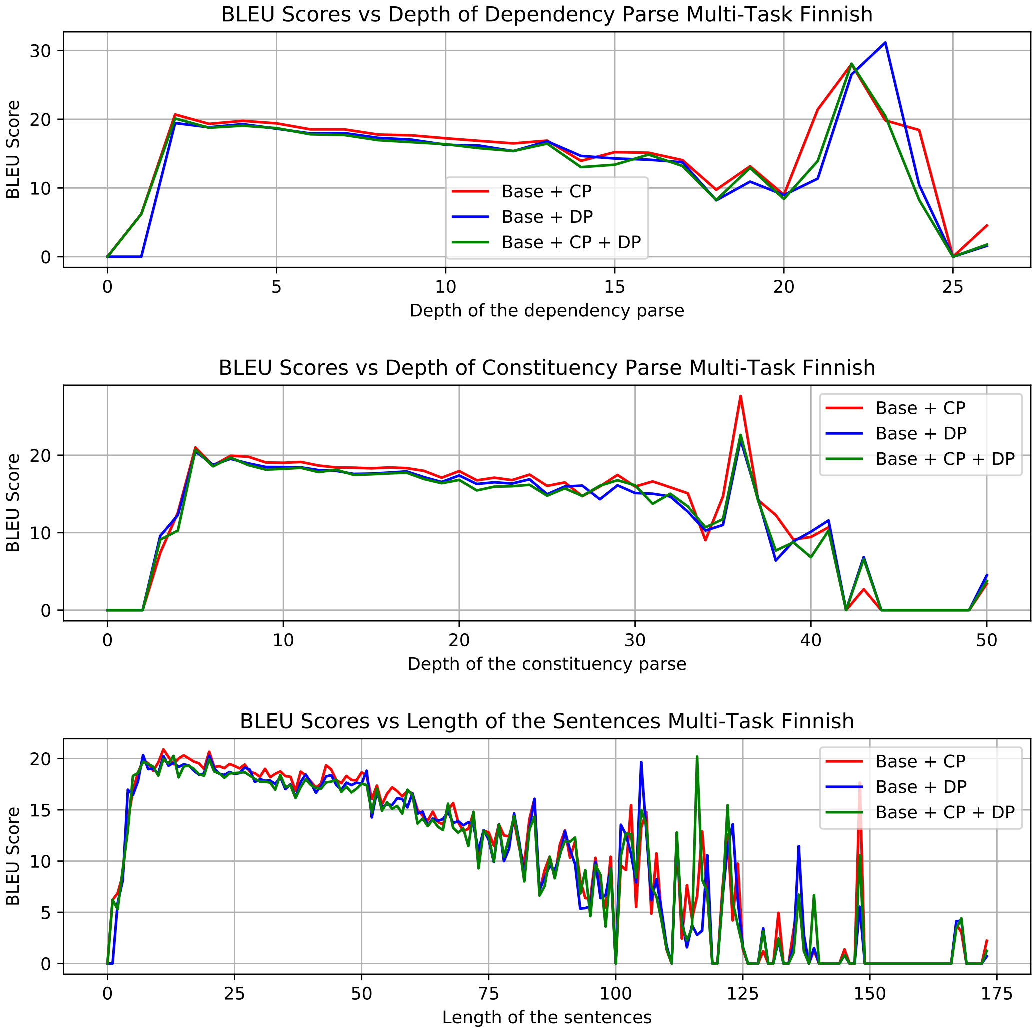


Figure 5: English-Finnish Multi-Task models performance over sentence complexities

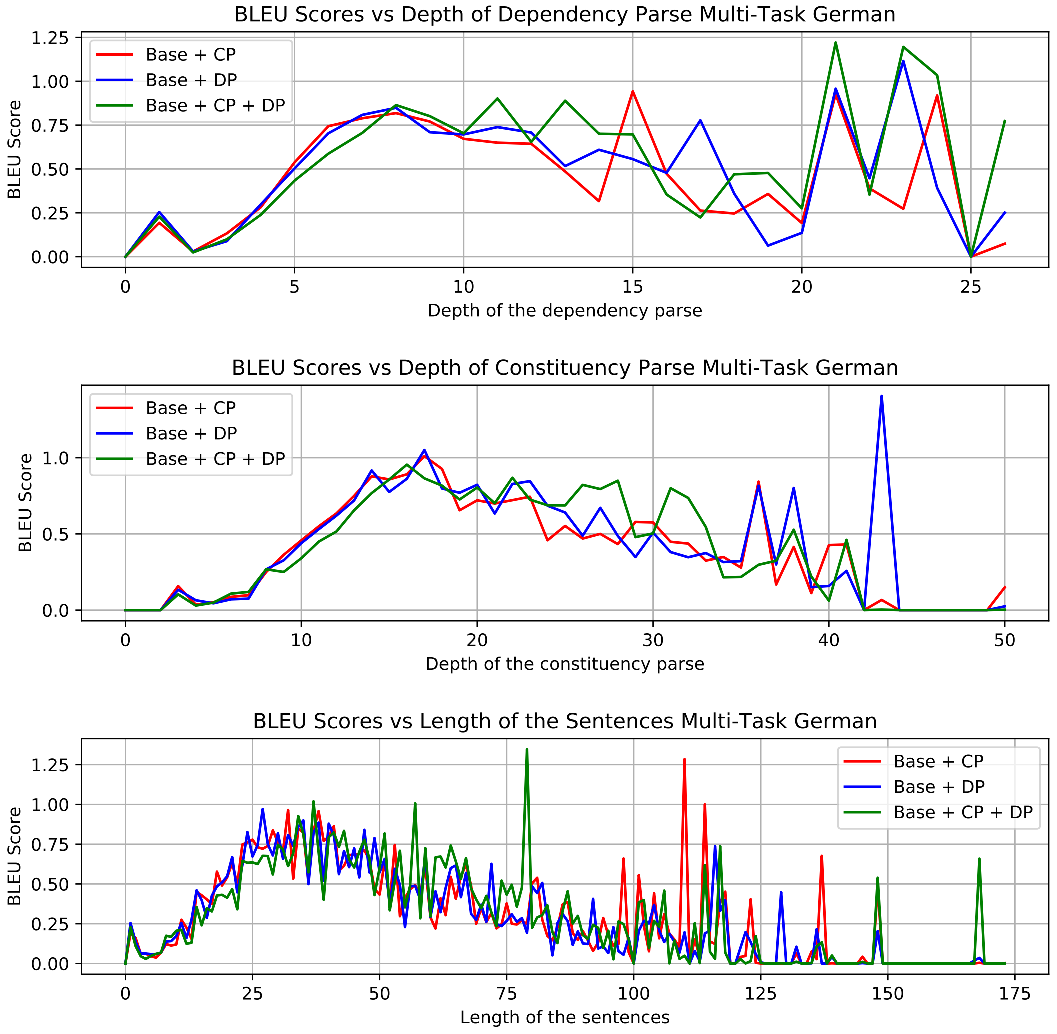


Figure 6: English-German Multi-Task models performance over sentence complexities

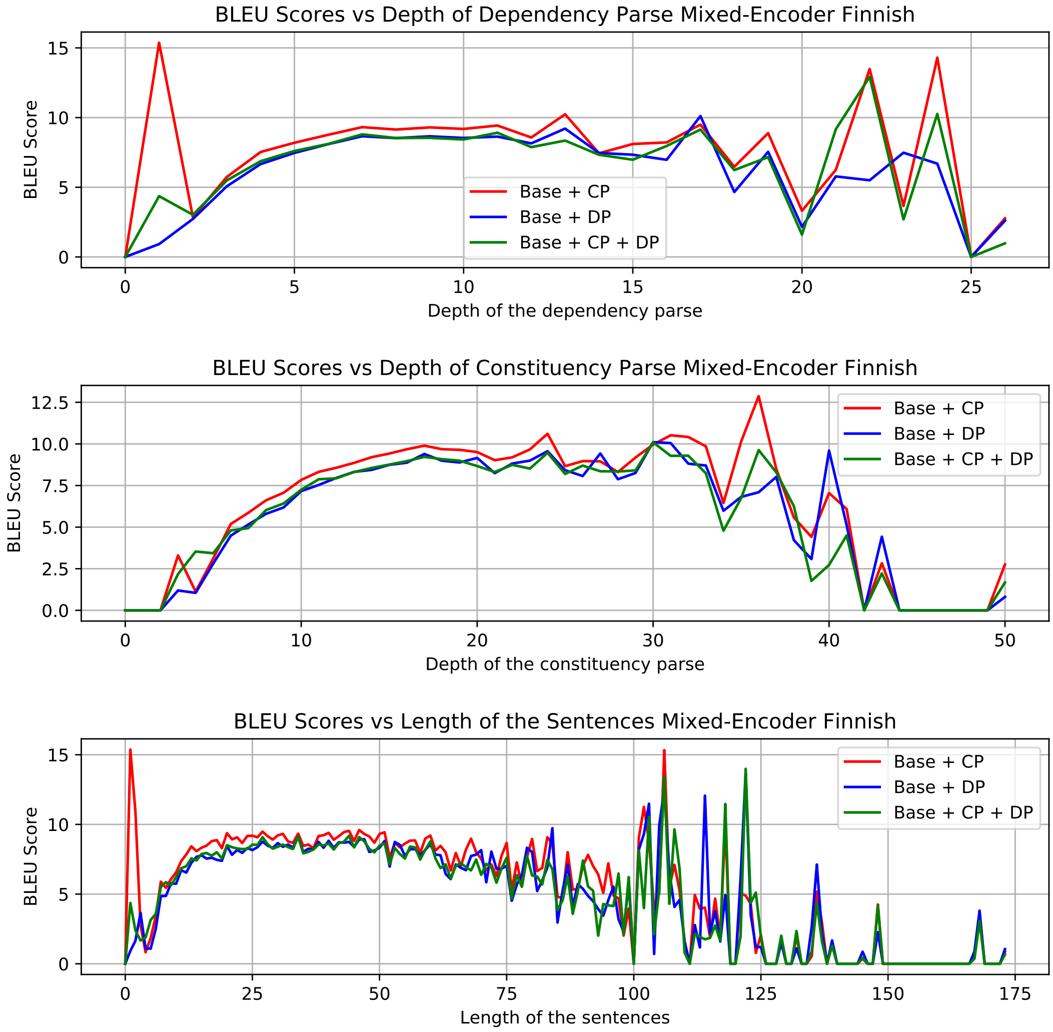


Figure 7: English-Finnish Mixed-Encoder models performance over sentence complexities

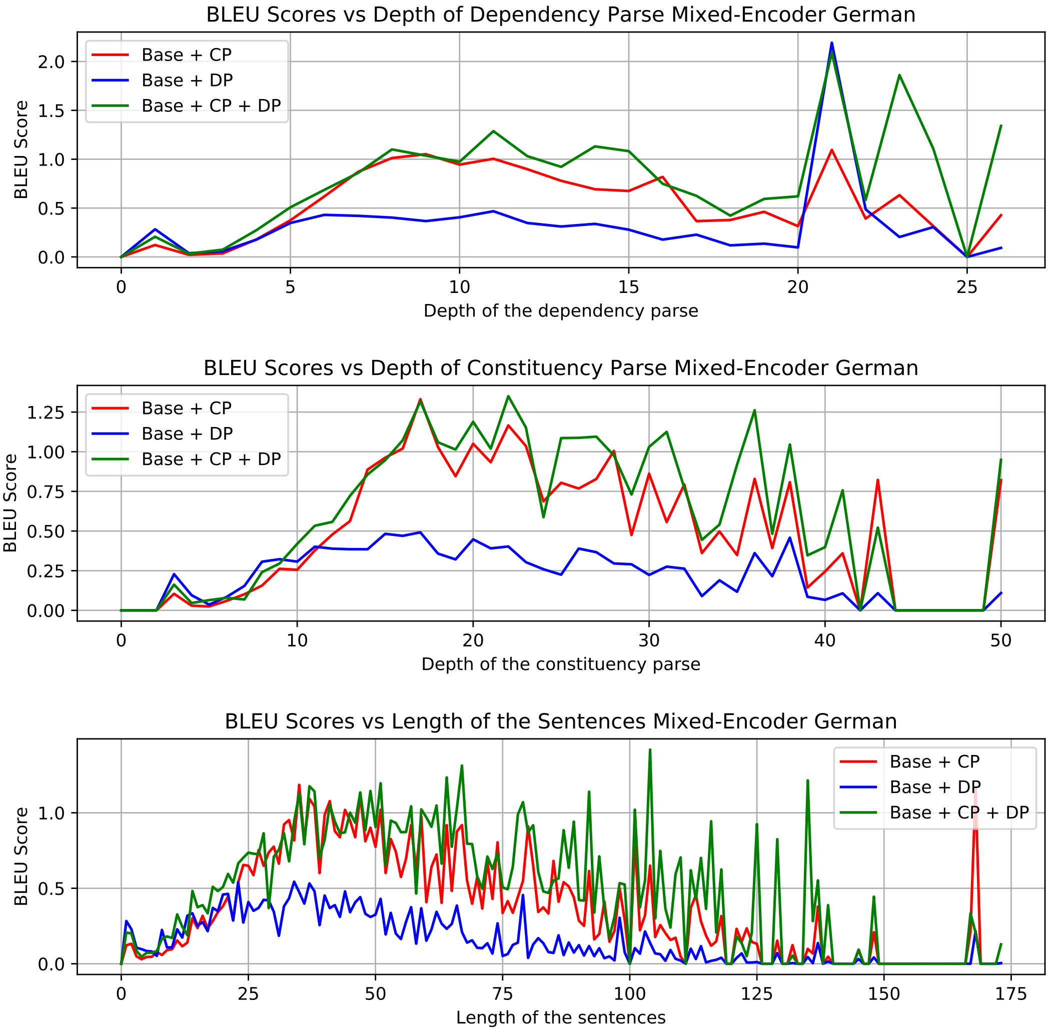


Figure 8: English-German Mixed-Encoder models performance over sentence complexities