Incorporating Source Syntax into Transformer-Based Neural Machine Translation

Anna Currey
University of Edinburgh
a.currey@sms.ed.ac.uk

Kenneth Heafield
University of Edinburgh
kheafiel@ed.ac.uk

Abstract

Transformer-based neural machine translation (NMT) has recently achieved state-of-the-art performance on many machine translation tasks. However, recent work (Raganato and Tiedemann, 2018; Tang et al., 2018; Tran et al., 2018) has indicated that Transformer models may not learn syntactic structures as well as their recurrent neural network-based counterparts, particularly in low-resource cases. In this paper, we incorporate constituency parse information into a Transformer NMT model. We leverage linearized parses of the source training sentences in order to inject syntax into the Transformer architecture without modifying it.

We introduce two methods: a multi-task machine translation and parsing model with a single encoder and decoder, and a mixed encoder model that learns to translate directly from parsed and unparsed source sentences. We evaluate our methods on low-resource translation from English into twenty target languages, showing consistent improvements of 1.3 BLEU on average across diverse target languages for the multi-task technique. We further evaluate the models on full-scale WMT tasks, finding that the multi-task model aids low- and medium-resource NMT but degenerates high-resource English→German translation.

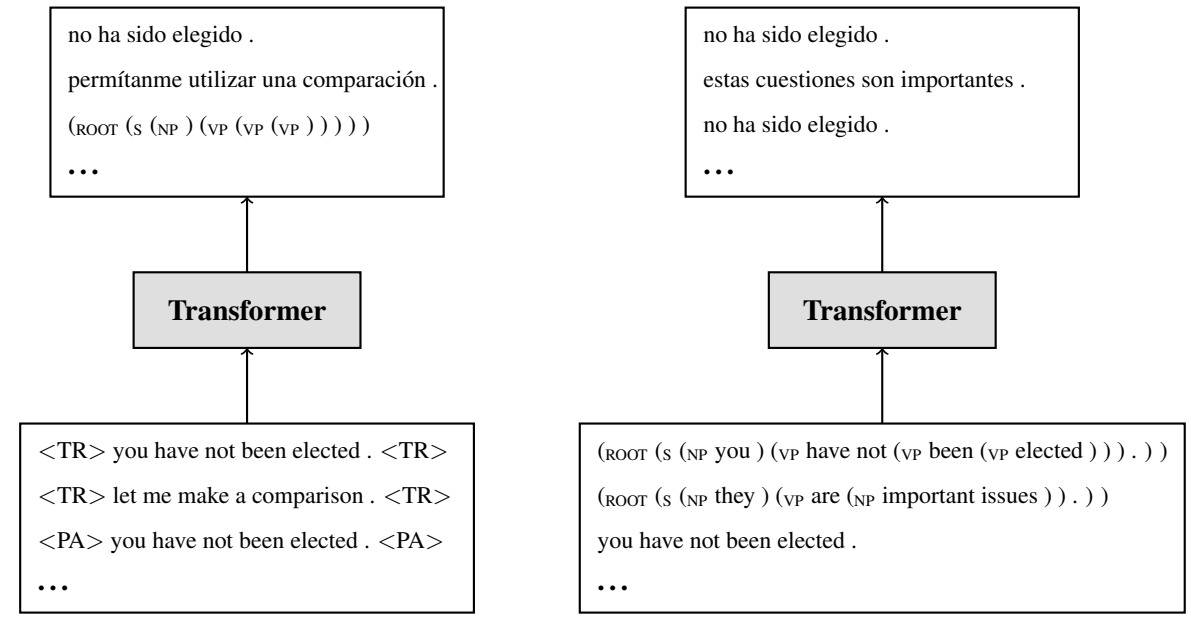
1 Introduction

Transformer-based neural machine translation (NMT) (Vaswani et al., 2017) has recently outperformed recurrent neural network (RNN)-based models (Bahdanau et al., 2015; Cho et al., 2014) in many tasks (Bojar et al., 2018). However, there is still room for improvement for NMT, particularly for low- and moderate-resource language pairs. Enriching NMT with syntactic information has the potential to improve generalization in low-resource scenarios, and adding syntax to Transformer-based NMT is currently an underexplored research area.

Transformer-based NMT may in fact stand to benefit even more from explicit syntactic annotations than RNN-based NMT, particularly in low-resource settings. On the one hand, the Transformer model already learns some syntax without explicit supervision in high-resource cases. Vaswani et al. (2017) visualized a few encoder self-attentions in a trained NMT model and found that they seemed to capture syntactic structure. This was formalized by Raganato and Tiedemann (2018), who found that Transformer encoders trained on high-resource NMT tasks were able to perform reasonably well at part-of-speech tagging, chunking, and other tasks. However, for Transformers trained on low-resource NMT, the results on these tasks were not as strong. Additionally, Tran et al. (2018) found that an RNN language model did better at predicting subject-verb agreement than a Transformer language model; Tang et al. (2018) saw similar results for Transformer vs. RNN NMT models.

Thus, the goal of this paper is to improve Transformer-based NMT using source-side syntactic supervision. We propose two methods that incorporate source-side linearized constituency parses into Transformer-based NMT. The first, multi-task, uses the Transformer to learn to parse and translate the source sentence simultaneously. The second, mixed encoder, learns to translate directly from both parsed and unparsed source sentences. This paper makes the following contributions:

- This is one of the first attempts at using syntax to improve Transformer-based NMT
- We introduce two methods for adding syntax



(a) Multi-task syntactic NMT model. The system is trained to translate (<TR>) and parse (<PA>) source sentences using the same architecture.

(b) Mixed encoder syntactic NMT model. The system learns to translate directly from both parsed and unparsed source sentences into unparsed target sentences.

Figure 1: Illustrations of the two proposed syntactic NMT methods.

to NMT that are straightforward to incorporate in practice

- We empirically evaluate both methods on translation from English into 21 diverse target languages, finding that the multi-task method improves consistently over a non-syntactic baseline

2 Transformer-Based NMT with Linearized Parses

We propose two models for incorporating linearized parses into Transformer-based NMT: a *multi-task* model and a *mixed encoder* model. Figure 1 summarizes the two proposed methods; they are discussed in detail in sections 2.2 and 2.3, respectively.

2.1 Linearized Constituency Parses

Both of our proposed methods make use of linearized parses of the source sentences to inject source syntax into Transformer-based NMT. Linearizing the parses allows us to add syntactic information without modifying the Transformer architecture. Here, we describe how these parses are created. We generate and format the parsed data as follows:

1. In order to generate syntactically parsed training data, we use the Stanford CoreNLP constituency parser (Manning et al., 2014) to parse the source side of the parallel corpus. This technique of parsing the parallel data instead of using gold parses is common in syntactic NMT (Eriguchi et al., 2016) and in neural parsing (Vinyals et al., 2015). For the multi-task model, it would be possible to incorporate gold parses into training as well, but we leave this for future work.

2. We linearize the resulting parses similarly to Vinyals et al. (2015) by using a depth-first tree traversal. We tokenize the opening parenthesis of each phrase with its phrase label.

3. Since neural machine translation already struggles with long sentences (Bahdanau et al., 2015), and adding the phrase nodes has the potential to make the sentences much longer, we remove part-of-speech tags from the parses (as was done by Aharoni and Goldberg, 2017).

4. For our multi-task model (section 2.2), we remove words from the linearized parses. We do this in order to further shorten the length of the target sequences. We do not expect that this will make the parsing task too difficult, as a similar technique was used for neural pars-

translation	<TR> you have not been elected . <TR> → no ha sido elegido .
parsing	<PA> you have not been elected . <PA> → (ROOT (S (NP) (VP (VP (VP)))))

Table 1: Example of English→Spanish training data for parsing and translation tasks in the multi-task system.

(ROOT (S (NP you) (VP have not (VP been (VP elected))) .)) → no ha sido elegido .
you have not been elected . → no ha sido elegido .

Table 2: Example of English→Spanish training data for the mixed encoder system.

ing by Vinyals et al. (2015).

5. For our mixed encoder model (section 2.3), we convert the words in the parses into subwords using byte pair encoding (Sennrich et al., 2016). We do not allow the parse labels to be broken into subwords.

Tables 1 and 2 give examples of the resulting parse formats.

2.2 Multi-Task NMT and Parsing with Shared Decoder

Our first method for incorporating source-side syntax into Transformer-based NMT adopts a multi-task framework. The main task is translating the source sentence into the target language; the secondary task is parsing the source sentence. For the parsing task, we employ the same encoder-decoder framework as for NMT, with the sequential source sentence as input and the linearized, unlexicalized parsed source sentence as output. Thus, both tasks are trained using a single model with a shared encoder and decoder. This is similar to the multi-task framework proposed by Luong et al. (2016), with three main differences: 1) we do not use separate decoders for each task, 2) we use the same source data for both parsing and translation, and 3) we use a Transformer rather than recurrent neural network-based architecture.

We do not directly use gold parses to train the parsing task, nor do we split the training data between the two tasks. The reason for using the same source data for both tasks is that we expect it to be difficult to find a sufficiently large amount of in-domain gold parses for training; additionally, our main goal is to improve NMT, so we do not expect the lower quality of the synthetic parses to matter.

In order to generate the training data for this model, we first create linearized parses of the source side of the training corpus as described above. Next, we add a tag at the beginning and end of each source sentence indicating the desired task, similar to what was done by Johnson et al. (2017) for multilingual NMT. Table 1 gives an example of the data format. Finally, we shuffle the parsing and translation training data together and train the shared encoder and decoder on both tasks, making no further distinction between the tasks during training. Since we parse all of the training data, each source sentence appears twice: once with a target language sentence and once with a parse of the source sentence. These copies are shuffled separately.

2.3 Mixed Encoder Transformer

Our second method for augmenting the NMT Transformer with syntax is the mixed encoder model. This model learns to translate both from unparsed and parsed source sentences into unparsed target sentences.

In order to train the mixed encoder model, we create two copies of the training data, one with parsed source sentences and the other with unparsed source sentences. We then shuffle these training corpora together into a single corpus and train a standard Transformer NMT model on the final data, with a single encoder for both parsed and unparsed source sentences. The training data contains (parsed source, unparsed target) and (unparsed source, unparsed target) sentence pairs; Table 2 gives an example of the two types of training sentence pairs for the mixed encoder method. Since the data is shuffled, these two sentence pairs (with identical target sentences) will not necessarily be seen together during training.

Since the mixed encoder model is trained on both parsed and unparsed source sentences, during inference it is able to translate from either source sentence format. Inference on unparsed source sentences is slightly faster (since it does not require parsing of the source sentence) and achieves slightly higher BLEU scores, so we show results using unparsed source sentences for our experiments (sections 4.2 and 5.2).

3 Experimental Setup

We evaluate our multi-task and mixed encoder models compared to a standard (non-syntactic) Transformer baseline on translation from English into 21 target languages. Sections 4.1 and 5.1 contain detailed information on the target languages and data used. All models are implemented in Sockeye (Hieber et al., 2017). For hyperparameter settings, we follow the recommendations of Vaswani et al. (2017).

We preprocess our data for all experiments as follows. First, we tokenize and truecase the data using the Moses scripts (Koehn et al., 2007). We then train separate subword vocabularies (Sennrich et al., 2016) for the source and target languages, with 30k merge operations per language. We use the Stanford CoreNLP parser (Manning et al., 2014) to generate constituency parses of the source (English) sentences, and linearize and format the parses as described in section 2.1. We do not use any monolingual training data; however, our proposed models are amenable to adding monolingual data, and we expect that BLEU scores would strongly increase if monolingual training data were used.

Family	Language	Abbrev.
Baltic	Latvian	LV
	Lithuanian	LT
Germanic	Danish	DA
	Dutch	NL
	German	DE
	Swedish	SV
Hellenic	Greek	EL
Romance	French	FR
	Italian	IT
	Portuguese	PT
	Romanian	RO
	Spanish	ES
Slavic	Bulgarian	BG
	Czech	CS
	Polish	PL
	Slovak	SK
	Slovene	SL
Uralic	Estonian	ET
	Finnish	FI
	Hungarian	HU

Table 3: Target languages used in our experiments, along with their language families or branches and their abbreviations (abbrev.).

4 Small-Scale Cross-Lingual Experiments

4.1 Data

We use the Europarl Parallel Corpus (Koehn, 2005) as the basis for our small-scale cross-lingual experiments. We consider translation from English (EN) into each of the twenty remaining target languages; Table 3 contains a full list of the target languages, as well as their language families or branches. By using this data set, we are able to evaluate the usefulness of syntactic information for several relatively diverse target languages, unlike most previous work on syntactic NMT (reviewed in section 7). However, all the languages in our experiments are Indo-European or Uralic due to using Europarl.

In order to facilitate comparison between the target languages, we follow Cotterell et al. (2018) by taking only the intersections of the Europarl training data. This means that the source (EN) data is identical for all experiments, and the targets are all translations of each other in the different target languages. This results in 170k parallel training sentences for each language pair. We reserve a random subset of 10k sentences from the original data to use as development data and an additional 10k sentences as test data; these development and test sets are not included in the training data.

4.2 Results

Table 4 displays BLEU scores on the test data for each target language for the proposed systems. The multi-task system outperforms the baseline for all target languages. In addition, for all but four target languages (SV, EL, SK, and ET), the multi-task system is at least 1 BLEU point better than the baseline. Thus, our proposed multi-task method consistently improves over a non-syntactic baseline across several diverse target languages in low-resource scenarios. Additionally, in all cases but two (EN→LT and EN→ET), multi-task achieves the highest BLEU score of all models.

The performance of the mixed encoder system in relation to the baseline is less consistent than that of the multi-task system. In most cases, the mixed encoder improves only slightly (less than 1 BLEU) over the baseline, although for LV, LT, RO, ES, PL, and FI, the improvements are stronger. However, for four target languages (NL, EL, BG,

EN→*	base	mixed enc.	multi-task
LV	26.5	28.1 (+1.6)	**28.2** (+1.7)
LT	23.5	24.6 (+1.1)	**24.8** (+1.3)
DA	39.5	40.1 (+0.6)	**40.7** (+1.2)
NL	28.8	28.7 (-0.1)	**30.6** (+1.8)
DE	30.5	30.6 (+0.1)	**32.1** (+1.6)
SV	35.9	**36.4** (+0.5)	**36.4** (+0.5)
EL	38.9	38.8 (-0.1)	**39.7** (+0.8)
FR	38.3	38.5 (+0.2)	**40.4** (+2.1)
IT	31.3	31.3 (==)	**32.5** (+1.2)
PT	39.2	39.3 (+0.1)	**40.5** (+1.3)
RO	36.3	**37.8** (+1.5)	**37.8** (+1.5)
ES	41.6	43.0 (+1.4)	**43.1** (+1.5)
BG	39.0	38.6 (-0.4)	**40.5** (+1.5)
CS	27.5	28.3 (+0.8)	**28.8** (+1.3)
PL	23.7	24.8 (+1.1)	**25.1** (+1.4)
SK	32.8	32.5 (-0.3)	**32.9** (+0.1)
SL	33.3	34.2 (+0.9)	**34.9** (+1.6)
ET	20.2	**20.9** (+0.7)	20.8 (+0.6)
FI	21.5	22.8 (+1.3)	**23.3** (+1.8)
HU	22.3	22.6 (+0.3)	**23.4** (+1.1)

Table 4: BLEU scores on the test set for small-scale cross-lingual experiments for the baseline (base), mixed encoder (mixed enc.), and multi-task models. Difference with the baseline is shown in parentheses.

and SK), the mixed encoder system does worse than the non-syntactic baseline.

Target language family does not seem to have a noticeable effect on the performance of either the mixed encoder or the multi-task method; this could be due to the fact that the syntactic annotations were on the source sentence only. It remains to be seen whether certain source languages are particularly amenable to incorporating source syntax in NMT.

5 Full-Scale WMT Experiments

5.1 Data

The main goal of the previous section was to evaluate our proposed syntactic NMT methods on a wide range of target languages and compare the effect of target language on performance. In this section, we run additional experiments in order to evaluate the proposed methods on a standard benchmark. We train our models on the following tasks: English→Turkish (TR) from the WMT18 news translation shared task (Bojar et al., 2018), English→Romanian WMT16 (Bojar et al., 2016), and English→German WMT17 (Bojar et al., 2017).

System	newstest2017	newstest2018
baseline	9.6	8.8
mixed enc.	9.6 (==)	9.3 (+0.5)
multi-task	**10.6** (+1.0)	**10.4** (+1.6)

Table 5: BLEU scores (and improvement over the baseline) for EN→TR on the test (newstest2017) and held-out (newstest2018) datasets.

For each experiment, we use all available parallel training data from the task, but no monolingual data. This gives us 200k parallel training sentences for EN→TR, 600k for EN→RO, and 5.9M for EN→DE. Note that the EN→RO and EN→DE training corpora contain some overlaps with the training data in section 4.1, although the experiments in this section use significantly more training data. We validate EN→TR on newstest2016, EN→RO on newsdev2016, and EN→DE on newstest2015.

5.2 Results

The results for the EN→TR experiments are displayed in Table 5. These results mirror what was seen in the previous experiments: the mixed encoder method gives modest improvements over the non-syntactic baseline (0–0.5 BLEU), while the multi-task method yields the strongest results, with an improvement of 1.0–1.6 BLEU points over the baseline. Although Turkish is not related to any of the target languages studied in section 4, the amount of training data for EN→TR is similar to what was used in the previous section, which might be one explanation for the similar results.

Table 6 shows performance of each model on the WMT EN→RO experiments. Here, we see more modest improvements from adding the syntactic data: only 0.5 BLEU over the baseline for both the mixed encoder and multi-task methods. It is interesting to compare this with the results for the Europarl EN→RO experiments (section 4.2); there, we saw a much larger improvement over the baseline for both multi-task models (1.5 BLEU). This indicates that the effectiveness of these models may depend on amount of data (the WMT models were trained on about three times as much training data) rather than on target language family.

Finally, we display our WMT EN→DE results in Table 7. Here, we see that for very high-resource EN→DE translation, the multi-task

System	newstest2016
baseline	21.5
mixed enc.	**22.0** (+0.5)
multi-task	**22.0** (+0.5)

Table 6: BLEU scores (and improvement over the baseline) for EN→RO on the test set (newstest2016).

System	newstest2016	newstest2017
baseline	31.7	25.5
mixed enc.	**31.9** (+0.2)	**26.0** (+0.5)
multi-task	29.6 (-2.1)	23.4 (-2.1)

Table 7: BLEU scores (and difference with the baseline) for EN→DE on the test (newstest2016) and held-out (newstest2017) datasets.

method does much worse than the baseline (by 2.1 BLEU points). In addition, the mixed encoder method achieves comparable BLEU scores to the baseline (only 0.2–0.5 BLEU higher). Thus, neither proposed technique is particularly successful for high-resource EN→DE NMT. Again, we can contrast this with the Europarl EN→DE experiments, where we saw strong improvements from the multi-task model (1.6 BLEU). This lends further credence to the hypothesis that these NMT models with linearized source parses are helpful cross-linguistically in low-resource scenarios, but not in high-resource setups.

We further investigated the WMT EN→DE multi-task model to find reasons for the large drop in performance compared to the baseline. We found that while the multi-task model was able to generate reasonable (albeit lower-quality) translations, it did not successfully learn to parse. During parsing inference, the model always output the same parse regardless of the input sentence: *(ROOT (S (NP) (VP (NP (NP) (PP (NP (NP) (PP (NP))))))))* . This was a common parse in the training data (it occurred 12k times in the data). This issue is partially due to the fact that validation is only done on the translation task, not on the parsing task. However, we do not see this issue with the other language pairs and experiments. This failure to learn to parse indicates that the WMT EN→DE multi-task model is not able to take advantage of the syntactic annotations.

6 Validity of Parses

The multi-task syntactic NMT models are trained both to translate and to parse the input sentences.

EN→*	% Valid Parses
LV	96.8%
LT	99.2%
DA	70.8%
NL	93.3%
DE	87.2%
SV	95.4%
EL	85.2%
FR	92.3%
IT	78.8%
PT	89.4%
RO	96.3%
ES	86.5%
BG	97.5%
CS	95.9%
PL	98.1%
SK	98.5%
SL	97.3%
ET	98.2%
FI	95.1%
HU	93.6%

Table 8: Percent of valid parses of the parses generated by the Europarl multi-task systems.

The main goal of these models has been to improve translation; those results were reported in sections 4.2 and 5.2. In this section, we analyze the validity of the parses produced by the multi-task systems. We use a standard parsing benchmark, WSJ section 23 of the Penn Treebank (Marcus et al., 1993), as the evaluation dataset in this section. We preprocess this dataset as described in section 3 before using it as the source data for the multi-task systems.

The multi-task models were trained to generate unlexicalized parses. Since we removed part-of-speech tags from the parses during preprocessing, it is not possible to automatically relexicalize the parses. This is because there is no one-to-one correspondence between the leaves of the parse tree and the number of words in the sentence. Thus, rather than evaluating the parses directly, we count the number of valid parses (i.e. parses with balanced parentheses) per target language.

Table 8 shows the percent of generated parses that were valid for the Europarl multi-task models. For most target languages, over 90% of the generated parses are valid.

Unlike for the translation results, target language family does seem to have an effect on the

EN→*	% Valid Parses
TR	86.3%
RO	99.8%
DE	100%

Table 9: Percent of valid parses of the parses generated by the WMT multi-task systems.

parsing results. Overall, Romance, Germanic, and Hellenic target language systems generate the fewest valid parses. This indicates that Baltic, Slavic, and Uralic target languages are most helpful in learning to parse English in a multi-task system. Thus, from our cross-lingual experiments, it seems that the parsing performance of a multi-task system depends on the target language, whereas we saw in the previous sections that the translation success depends more on the amount of training data. Note, however, some caveats: 1) we did not perform validation on the parsing task (only on the translation task), and 2) we are measuring only parsing validity here, rather than parsing performance.

Table 9 shows the percent of valid parses for the three WMT multi-task experiments. For EN→DE, all of the generated parses are valid because they are all identical (as dicussed in section 5.2). For EN→RO, nearly all the parses are valid as well. However, this language pair did not have the same issue as EN→DE: the parses generated for each sentence were different, and a manual analysis indicated that the generated EN→RO parses were reasonable. The EN→TR system generated a large amount of valid parses, but fewer than the EN→RO system; it is possible that the EN→TR system would have done better with more training data.

7 Related Work

The performance of many RNN-based NMT paradigms has been improved by adding explicit syntactic annotations, particularly on the source side; we review some syntactic NMT models here. This paper is, along with Wu et al. (2018) and Zhang et al. (2019), among the first to add explicit syntax to Transformer-based NMT.

7.1 Linearized Parses in Neural Networks

In this work, we use linearized parse trees to add syntax into the Transformer. Vinyals et al. (2015) and Choe and Charniak (2016) introduced the idea of linearizing parse trees for neural parsing. Linearized parses are advantageous because they can be used anywhere that standard sequences can be used; in fact, Vaswani et al. (2017) showed that they can also be used by the Transformer to learn constituency parsing. Here, we leverage this idea by using linearized parses as an additional signal for the Transformer during NMT training.

7.2 Syntactic NMT with Modified Encoder

There have been several recent proposals to incorporate source-side syntax into RNN-based NMT by modifying the encoder architecture; we review some such models here. Eriguchi et al. (2016) augmented the RNN encoder with a tree-LSTM (Tai et al., 2015) to read in source-side HPSG parses, and combined this with a standard RNN decoder. Similarly, Bastings et al. (2017) used a graph convolutional encoder in combination with an RNN decoder to translate from dependency parsed source sentences. Although these models improved over non-syntactic RNN-based NMT systems, they relied heavily on parsed data during both training and inference, whereas our models are able to translate unparsed data. In addition, it is not clear how to incorporate such improvements into the state-of-the-art Transformer architecture.

7.3 Linearized Parses in NMT

This work fits with another line of research that uses linearized parses to incorporate syntax into neural machine translation without requiring a specific NMT architecture. Luong et al. (2016) used a single encoder and different decoders to train two tasks: parsing the source sentence and translating from source to target. Kiperwasser and Ballesteros (2018) also applied multi-task learning to syntactic NMT; they used a shared RNN decoder for translation, dependency parsing, and part-of-speech tagging and evaluated different scheduling techniques to combine the tasks. Our multi-task system builds off these two papers by training a joint NMT and parsing model using a single encoder and decoder in a Transformer framework, and further evaluates the multi-task framework on several language pairs.

Currey and Heafield (2018) leveraged a multi-source NMT system to learn to translate from both unparsed and parsed source sentences. Wu et al. (2018) similarly combined the standard bidirectional encoder with two additional encoders, one

that encoded the pre-order traversal of the dependency parse of the sentence and one that encoded the post-order traversal. Unlike Currey and Heafield (2018), they joined the encoders on the word level and used a Transformer architecture. Our mixed encoder model is similar to these but instead uses a single Transformer encoder for both parsed and unparsed source sentences.

The mixed RNN encoder model of Li et al. (2017) is also similar to our mixed encoder model; their model used an RNN to encode a linearized parse of a source sentence, but attended only to the words of the parse. Our mixed encoder model is trained on both linearized parses and unparsed sentences, but for the linearized parses we attend to words and to parse labels. Zhang et al. (2019) used syntax to augment the word representations in both RNN-based and Transformer-based NMT; this was done by concatenating the hidden states of a dependency parser with the NMT word embeddings. Their method is complementary to ours and could be used along with our multi-task or mixed encoder models to enhance any NMT architecture.

In this work, we have concentrated on source-side syntax, but linearized parses have also been popular for incorporating target syntax into neural machine translation. Aharoni and Goldberg (2017) and Nadejde et al. (2017) both trained RNN-based neural machine translation systems to translate from sequential source sentences into linearized parses of target sentences; this could also be done using a Transformer.

8 Conclusions

In this paper, we proposed two methods for incorporating source-side syntactic annotations into a Transformer-based neural machine translation system. The first, multi-task, used a shared encoder and decoder to train two tasks: translation and constituency parsing. The second, mixed encoder, learned to translate linearized parses of the source sentences as well as unparsed source sentences directly into the target language. We performed experiments from English into twenty target languages in a low-resource setup; the multi-task system improved over the non-syntactic baseline for all target languages. We further demonstrated the success of this method on the EN→TR and EN→RO WMT datasets; however, for the very high-resource EN→DE WMT setup, the multi-task model performed poorly, while the mixed encoder model did only marginally better than the non-syntactic baseline.

In the future, we plan on extending these techniques to incorporate target-side syntax into Transformer-based NMT. In addition, we would like to experiment with different source languages in order to find out whether adding source-side syntax has a greater effect on some source languages than others. It would also be interesting to experiment with a multi-task, multilingual NMT framework with multiple target languages.

References

Roee Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the ACL*, pages 132–140. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 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.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of the Third Conference on Machine Trans-*

lation, pages 272–303. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734. Association for Computational Linguistics.

Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336. Association for Computational Linguistics.

Ryan Cotterell, Sebastian J. Mielke, Jason Eisner, and Brian Roark. 2018. Are all languages equally hard to language-model? In *Proceedings of NAACL-HLT*, pages 536–541. Association for Computational Linguistics.

Anna Currey and Kenneth Heafield. 2018. Multi-source syntactic neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2961–2966. Association for Computational Linguistics.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 823–833. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Eliyahu Kiperwasser and Miguel Ballesteros. 2018. Scheduled multi-task learning: From syntax to translation. *Transactions of the Association for Computational Linguistics*, 6:225–240.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *10th Machine Translation Summit*, volume 5, pages 79–86.

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*, pages 177–180. Association for Computational Linguistics.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017. Modeling source syntax for neural machine translation. In *Proceedings of the 55th Annual Meeting of the ACL*, pages 688–697. Association for Computational Linguistics.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the ACL*, pages 55–60. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Maria Nadejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. 2017. Predicting target language CCG supertags improves neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 68–79. 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. Association for Computational Linguistics.

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 ACL*, pages 1715–1725. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing*, pages 1556–1566. Association for Computational Linguistics.

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. 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. Association for Computational Linguistics.

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.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems 28*, pages 2773–2781.

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*, 26(11):2132–2141.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2019. Syntax-enhanced neural machine translation with syntax-aware word representations. In *Proceedings of NAACL*, pages 1151–1161. Association for Computational Linguistics.