Predicting Target Language CCG Supertags Improves Neural Machine Translation

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Accepted author manuscript

Published In:
Proceedings of the Conference on Machine Translation (WMT), Volume 1: Research Papers

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Predicting Target Language CCG Supertags Improves Neural Machine Translation

Maria N˘adejde¹ and Siva Reddy¹ and Rico Sennrich¹ and Tomasz Dwojak¹,²
Marcin Junczys-Dowmunt² and Philipp Koehn³ and Alexandra Birch¹
¹School of Informatics, University of Edinburgh
²Adam Mickiewicz University
³Dep. of Computer Science, Johns Hopkins University
{m.nadejde,siva.reddy, rico.sennrich, a.birch}@ed.ac.uk
{t.dwojak,junczys}@amu.edu.pl, phi@jhu.edu

Abstract

Neural machine translation (NMT) models are able to partially learn syntactic information from sequential lexical information. Still, some complex syntactic phenomena such as prepositional phrase attachment are poorly modeled. This work aims to answer two questions: 1) Does explicitly modeling target language syntax help NMT? 2) Is tight integration of words and syntax better than multitask training? We introduce syntactic information in the form of CCG supertags in the decoder, by interleaving the target supertags with the word sequence. Our results on WMT data show that explicitly modeling target-syntax improves machine translation quality for German→English, a high-resource pair, and for Romanian→English, a low-resource pair and also several syntactic phenomena including prepositional phrase attachment. Furthermore, a tight coupling of words and syntax improves translation quality more than multitask training. By combining target-syntax with adding source-side dependency labels in the embedding layer, we obtain a total improvement of 0.9 BLEU for German→English and 1.2 BLEU for Romanian→English.

1 Introduction

Sequence-to-sequence neural machine translation (NMT) models (Sutskever et al., 2014; Cho et al., 2014b; Bahdanau et al., 2015) are state-of-the-art on a multitude of language-pairs (Sennrich et al., 2016a; Junczys-Dowmunt et al., 2016). Part of the appeal of neural models is that they can learn to implicitly model phenomena which underlie high quality output, and some syntax is indeed captured by these models. In a detailed analysis, Bentivogli et al. (2016) show that NMT significantly improves over phrase-based SMT, in particular with respect to morphology and word order, but that results can still be improved for longer sentences and complex syntactic phenomena such as prepositional phrase (PP) attachment. Another study by Shi et al. (2016) shows that the encoder layer of NMT partially learns syntactic information about the source language, however complex syntactic phenomena such as coordination or PP attachment are poorly modeled.

Recent work which incorporates additional source-side linguistic information in NMT models (Luong et al., 2016; Sennrich and Haddow, 2016) show that even though neural models have strong learning capabilities, explicit features can still improve translation quality. In this work, we examine the benefit of incorporating global syntactic information on the target-side. We also address the question of how best to incorporate this information. For language pairs where syntactic resources are available on both the source and target-side, we show that approaches to incorporate source syntax and target syntax are complementary.

We propose a method for tightly coupling words and syntax by interleaving the target syntactic representation with the word sequence. We compare this to loosely coupling words and syntax using a multitask solution, where the shared parts of the model are trained to produce either a target sequence of words or supertags in a similar fashion to Luong et al. (2016).

We use CCG syntactic categories (Steedman, 2000), also known as supertags, to represent syntax explicitly. Supertags provide global syntactic information locally at the lexical level. They encode subcategorization information, capturing short and long range dependencies and attach-
ments, and also tense and morphological aspects of the word in a given context. Consider the sentence in Figure 1. This sentence contains two PP attachments and could lead to several disambiguation possibilities (“in” can attach to “Netanyahu” or “receives”, and “of” can attach to “capital”, “Netanyahu” or “receives”). These alternatives may lead to different translations in other languages. However the supertag \((S[dcl]\backslash NP)/PP)/NP\) of “receives” indicates that the preposition “in” attaches to the verb, and the supertag \((NP\backslash NP)/NP\) of “of” indicates that it attaches to “capital”, thereby resolving the ambiguity.

Our research contributions are as follows:

- We propose a novel approach to integrating target syntax at word level in the decoder, by interleaving CCG supertags in the target word sequence.
- We show that the target language syntax improves translation quality for German→English and Romanian→English as measured by BLEU. Our results suggest that a tight coupling of target words and syntax (by interleaving) improves translation quality more than the decoupled signal from multitask training.
- We show that incorporating source-side linguistic information is complimentary to our method, further improving the translation quality.
- We present a fine-grained analysis of SNMT and show consistent gains for different linguistic phenomena and sentence lengths.

2 Related work

Syntax has helped in statistical machine translation (SMT) to capture dependencies between distant words that impact morphological agreement, subcategorisation and word order (Galley et al., 2004; Menezes and Quirk, 2007; Williams and Koehn, 2012; Nadejde et al., 2013; Sennrich, 2015; Nadejde et al., 2016a,b; Chiang, 2007). There has been some work in NMT on modeling source-side syntax implicitly or explicitly. Kalchbrenner and Blunsom (2013); Cho et al. (2014a) capture the hierarchical aspects of language implicitly by using convolutional neural networks, while Eriguchi et al. (2016) use the parse tree of the source sentence to guide the recurrence and attention model in tree-to-sequence NMT. Luong et al. (2016) co-train a translation model and a source-side syntactic parser which share the encoder. Our multitask models extend their work to attention-based NMT models and to predicting target-side syntax as the secondary task. Sennrich and Haddow (2016) generalize the embedding layer of NMT to include explicit linguistic features such as dependency relations and part-of-speech tags and we use their framework to show source and target syntax provide complementary information.

Applying more tightly coupled linguistic factors on the target for NMT has been previously investigated. Niehues et al. (2016) proposed a factored RNN-based language model for re-scoring an n-best list produced by a phrase-based MT system. In recent work, Martínez et al. (2016) implemented a factored NMT decoder which generated both lemmas and morphological tags. The two factors were then post-processed to generate the word form. Unfortunately no real gain was reported for these experiments. Concurrently with our work, Aharoni and Goldberg (2017) proposed serializing the target constituency trees and Eriguchi et al. (2017) model target dependency relations by augmenting the NMT decoder with a RNN grammar (Dyer et al., 2016). In our work, we use CCG supertags which are a more compact representation of global syntax. Furthermore, we do not focus on model architectures, and instead we explore the more general problem of including target syntax in NMT: comparing tightly and loosely coupled syntactic information and showing source and target syntax are complementary.

Previous work on integrating CCG supertags in factored phrase-based models (Birch et al., 2007) made strong independence assumptions between the target word sequence and the CCG categories. In this work we take advantage of the expressive power of recurrent neural networks to learn representations that generate both words and CCG supertags, conditioned on the entire lexical and syntactic target history.

3 Modeling Syntax in NMT

CCG is a lexicalised formalism in which words are assigned with syntactic categories, i.e., supertags, that indicate context-sensitive morpho-syntactic properties of a word in a sentence. The combinators of CCG allow the supertags to capture global syntactic constraints locally. Though NMT
captures long range dependencies using long-term memory, short-term memory is cheap and reliable. Supertags can help by allowing the model to rely more on local information (short-term) and not having to rely heavily on long-term memory.

Consider a decoder that has to generate the following sentences:

1. What \((S[wq]/(S[q]/NP))/N\) city is \((S[q]/PP)/NP\) the Taj Mahal in?

2. Where \((S[wq]/(S[q]/NP)) is\( (S[q]/NP)/NP\) the Taj Mahal?

If the decoding starts with predicting “What”, it is ungrammatical to omit the preposition “in”, and if the decoding starts with predicting “Where”, it is ungrammatical to predict the preposition. Here the decision to predict “in” depends on the first word, a long range dependency. However if we rely on CCG supertags, the supertags of both these sequences look very different. The supertag \((S[q]/PP)/NP\) for the verb “is” in the first sentence indicates that a preposition is expected in future context. Furthermore it is likely to see this particular supertag of the verb in the context of \((S[wq]/(S[q]/NP))/N\) but it is unlikely in the context of \((S[wq]/(S[q]/NP))\). Therefore a succession of local decisions based on CCG supertags will result in the correct prediction of the preposition in the first sentence, and omitting the preposition in the second sentence. Since the vocabulary of CCG supertags is much smaller than that of possible words, the NMT model will do a better job at generalizing over and predicting the correct CCG supertags sequence.

CCG supertags also help during encoding if they are given in the input, as we saw with the case of PP attachment in Figure 1. Translation of the correct verb form and agreement can be improved with CCG since supertags also encode tense, morphology and agreements. For example, in the sentence “It is going to rain”, the supertag \((S[ng]\\NP[expl]]/(S[to]\\NP)\) of “going” indicates the current word is a verb in continuous form looking for an infinitive construction on the right, and an expletive pronoun on the left.

We explore the effect of target-side syntax by using CCG supertags in the decoder and by combining these with source-side syntax in the encoder, as follows.

**Baseline decoder** The baseline decoder architecture is a conditional GRU with attention \((cGRU_{attn})\) as implemented in the Nematus toolkit (Sennrich et al., 2017). The decoder is a recursive function computing a hidden state \(s_j\) at each time step \(j \in [1, T]\) of the target recurrence. This function takes as input the previous hidden state \(s_{j-1}\), the embedding of the previous target word \(y_{j-1}\) and the output of the attention model \(c_j\). The attention model computes a weighted sum over the hidden states \(h_i = [\hat{h}_i; \tilde{h}_i]\) of the bidirectional RNN encoder. The function \(g\) computes the intermediate representation \(t_j\) and passes this to a softmax layer which first applies a linear transformation \((W_o)\) and then computes the probability distribution over the target vocabulary. The training objective for the entire architecture is minimizing the discrete cross-entropy, therefore the loss \(l\) is the negative log-probability of the reference sentence.

\[
s_j' = GRU_1(y_{j-1}, s_{j-1}) \tag{1}
\]
\[
c_j = ATT([h_1; ...; h_{|x|}], s_j') \tag{2}
\]
\[
s_j = cGRU_{attn}(y_{j-1}, s_{j-1}, c_j) \tag{3}
\]
\[
t_j = g(y_{j-1}, s_j, c_j) \tag{4}
\]
\[
p_y = \prod_{j=1}^{T} p(y_j | x, y_{1:j-1}) = \prod_{j=1}^{T} \text{softmax}(t_j W_o) \tag{5}
\]
\[
l = -\log(p_y) \tag{6}
\]

**Target-side syntax** When modeling the target-side syntactic information we consider different
strategies of coupling the CCG supertags with the translated words in the decoder: interleaving and multitasking with shared encoder. In Figure 2 we represent graphically the differences between the two strategies and in the next paragraphs we formalize them.

- **Interleaving** In this paper we propose a tight integration in the decoder of the syntactic representation and the surface forms. Before each word of the target sequence we include its supertag as an extra token. The new target sequence $y'$ will have the length $2T$, where $T$ is the number of target words. With this representation, a single decoder learns to predict both the target supertags and the target words conditioned on previous syntactic and lexical context. We do not make changes to the baseline NMT decoder architecture, keeping equations (1) - (6) and the corresponding set of parameters unchanged. Instead, we augment the target vocabulary to include both words and CCG supertags. This results in a shared embedding space and the following probability of the target sequence $y'$, where $y'_j$ can be either a word or a tag:

$$y' = y_{1:1}^{tag}, y_{1:1}^{word}, ..., y_{T:1}^{tag}, y_{T:1}^{word}$$

$$p_{y'} = \prod_j p(y'_j | x, y'_{1:j-1})$$

At training time we pre-process the target sequence to add the syntactic annotation and then split only the words into byte-pair-encoding (BPE) (Sennrich et al., 2016b) sub-units. At testing time we delete the predicted CCG supertags to obtain the final translation. Figure 1 gives an example of the target-side representation in the case of interleaving. The supertag NP corresponding to the word *Netanyahu* is included only once before the three BPE subunits *Net+ an+ yahu*.

- **Multitasking – shared encoder** A loose coupling of the syntactic representation and the surface forms can be achieved by co-training a translation model with a secondary prediction task, in our case CCG supertagging. In the multitask framework (Luong et al., 2016) the encoder part is shared while the decoder is different for each of the prediction tasks: translation and tagging. In contrast to Luong et al., we train a separate attention model for each task and perform multitask learning with target syntax. The two decoders take as input the same source context, represented by the encoder’s hidden states $h_i = [h_i^\rightarrow; h_i^\leftarrow]$. However, each task has its own set of parameters associated with the five components of the decoder: $\text{GRU}_1$, $\text{ATT}$, $\text{cGRU}_{att}$, $g$, $\text{softmax}$. Furthermore, the two decoders may predict a different number of target symbols, resulting in target sequences of different lengths $T_1$ and $T_2$. This results in two probability distributions over separate target vocabularies for the words and the tags:

$$p_{y'}^\text{word} = \prod_j p(y'_j^{\text{word}} | x, y'_{1:j-1}^{\text{word}})$$

$$p_{y'}^\text{tag} = \prod_k p(y'_k^{\text{tag}} | x, y'_{1:k-1}^{\text{tag}})$$

Figure 2: Integrating target syntax in the NMT decoder: a) interleaving and b) multitasking.
The final loss is the sum of the losses for the two decoders:

\[ l = -\log(p_{\text{word}}^y) + \log(p_{\text{tag}}^y) \]  

(11)

We use EasySRL to label the English side of the parallel corpus with CCG supertags\(^1\) instead of using a corpus with gold annotations as in Luong et al. (2016).

**Source-side syntax – shared embedding** While our focus is on target-side syntax, we also experiment with including source-side syntax to show that the two approaches are complementary.

Sennrich and Haddow propose a framework for including source-side syntax as extra features in the NMT encoder. They extend the model of Bahdanau et al. by learning a separate embedding for several source-side features such as the word itself or its part-of-speech. All feature embeddings are concatenated into one embedding vector which is used in all parts of the encoder model instead of the word embedding. When modeling the source-side syntactic information, we include the CCG supertags or dependency labels as extra features. The baseline features are the subword units obtained using BPE together with the annotation of the subword structure using IOB format by marking if a symbol in the text forms the beginning (B), inside (I), or end (E) of a word. A separate tag (O) is used if a symbol corresponds to the full word. The word level supertag is replicated for each BPE unit. Figure 1 gives an example of the source-side feature representation.

### 4 Experimental Setup and Evaluation

#### 4.1 Data and methods

We train the neural MT systems on all the parallel data available at WMT16 (Bojar et al., 2016) for the German↔English and Romanian↔English language pairs. The English side of the training data is annotated with CCG lexical tags\(^2\) using EasySRL (Lewis et al., 2015) and the available pre-trained model\(^3\). Some longer sentences cannot be processed by the parser and therefore we eliminate them from our training and test data. We report the sentence counts for the filtered data sets in Table 1. Dependency labels are annotated with ParZU (Sennrich et al., 2013) for German and SyntaxNet (Andor et al., 2016) for Romanian.

All the neural MT systems are attentional encoder-decoder networks (Bahdanau et al., 2015) as implemented in the Nematus toolkit (Sennrich et al., 2017).\(^4\) We use similar hyper-parameters to those reported by (Sennrich et al., 2016a; Sennrich and Haddow, 2016) with minor modifications: we used mini-batches of size 60 and Adam optimizer (Kingma and Ba, 2014). We select the best single models according to BLEU on the development set and use the four best single models for the ensembles.

To show that we report results over strong baselines, table 2 compares the scores obtained by our baseline system to the ones reported in Sennrich et al. (2016a). We normalize diacritics\(^5\) for the English→Romanian test set. We did not remove or normalize Romanian diacritics for the other experiments reported in this paper. Our baseline systems are generally stronger than Sennrich et al. (2016a) due to training with a different optimizer for more iterations.

<table>
<thead>
<tr>
<th></th>
<th>This work</th>
<th>Sennrich et. al</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE→EN</td>
<td>31.0</td>
<td>28.5</td>
</tr>
<tr>
<td>EN→DE</td>
<td>27.8</td>
<td>26.8</td>
</tr>
<tr>
<td>RO→EN</td>
<td>28.0</td>
<td>27.8</td>
</tr>
<tr>
<td>EN→RO(^1)</td>
<td>25.6</td>
<td>23.9</td>
</tr>
</tbody>
</table>

Table 2: Comparison of baseline systems in this work and in Sennrich et al. (2016a). Case-sensitive BLEU scores reported over news.test2016 with mteval-13a.perl. \(^1\)Normalized diacritics.

During training we validate our models with BLEU (Papineni et al., 2002) on development sets: newstest2013 for German↔English and newstestdev2016 for Romanian↔English. We evaluate the systems on newstest2016 test sets for both lan-
language pairs and use bootstrap resampling (Riezler and Maxwell, 2005) to test statistical significance. We compute BLEU with multi-bleu.perl over tokenized sentences both on the development sets, for early stopping, and on the test sets for evaluating our systems.

Words are segmented into sub-units that are learned jointly for source and target using BPE (Sennrich et al., 2016b), resulting in a vocabulary size of 85,000. The vocabulary size for CCG supertags was 500.

For the experiments with source-side features we use the BPE sub-units and the IOB tags as baseline features. We keep the total word embedding size fixed to 500 dimensions. We allocate 10 dimensions for dependency labels when using these as source-side features and when using source-side CCG supertags we allocate 135 dimensions.

The interleaving approach to integrating target syntax increases the length of the target sequence. Therefore, at training time, when adding the CCG supertags in the target sequence we increase the maximum length of sentences from 50 to 100. On average, the length of English sentences for newstest2013 in BPE representation is 22.7, while the average length when adding the CCG supertags is 44. Increasing the length of the target recurrence results in larger memory consumption and slower training. Roughly 10h30 per 100,000 sentences (20,000 batches) for SNMT compared to 6h for NMT.

4.2 Results

In this section, we first evaluate the syntax-aware NMT model (SNMT) with target-side CCG supertags as compared to the baseline NMT model described in the previous section (Bahdanau et al., 2015; Sennrich et al., 2016a). We show that our proposed method for tightly coupling target syntax via interleaving, improves translation for both German→English and Romanian→English while the multitasking framework does not. Next, we show that SNMT with target-side CCG supertags can be complemented with source-side dependencies, and that combining both types of syntax brings the most improvement. Finally, our experiments with source-side CCG supertags confirm that global syntax can improve translation either as extra information in the encoder or in the decoder.

Target-side syntax We first evaluate the impact of target-side CCG supertags on overall translation quality. In Table 3 we report results for German→English, a high-resource language pair, and for Romanian→English, a low-resource language pair. We report BLEU scores for both the best single models and ensemble models. However, we will only refer to the results with ensemble models since these are generally better.

The SNMT system with target-side syntax improves BLEU scores by 0.9 for Romanian→English and by 0.6 for German→English. Although the training data for German→English is large, the CCG supertags still improve translation quality. These results suggest that the baseline NMT decoder benefits from modeling the global syntactic information locally via supertags.

Next, we evaluate whether there is a benefit to tight coupling between the target word sequence and syntax, as opposed to loose coupling. We compare our method of interleaving the CCG supertags with multitasking, which predicts target CCG supertags as a secondary task. The results in Table 3 show that the multitask approach does not improve BLEU scores for German→English, which exhibits long distance word reordering. For Romanian→English, which exhibits more local word reordering, multitasking improves BLEU by 0.6 relative to the baseline. In contrast, the interleaving approach improves translation quality for both language pairs and to a larger extent. Therefore, we conclude that a tight integration of the target syntax and word sequence is important. Conditioning the prediction of words on their corresponding CCG supertags is what sets SNMT apart from the multitasking approach.

Source-side and target-side syntax We now show that our method for integrating target-side syntax can be combined with the framework of Sennrich and Haddow (2016) for integrating source-side linguistic information, leading to further improvement in translation quality. We evaluate the syntax-aware NMT system, with CCG supertags as target-syntax and dependency labels as source-syntax. While the dependency labels do not encode global syntactic information, they disambiguate the grammatical function of words. Ini-
German→English & Romanian→English &

<table>
<thead>
<tr>
<th>model</th>
<th>syntax</th>
<th>strategy</th>
<th>single</th>
<th>ensemble</th>
<th>single</th>
<th>ensemble</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMT</td>
<td>-</td>
<td>-</td>
<td>31.0</td>
<td>31.2</td>
<td>28.1</td>
<td>28.4</td>
</tr>
<tr>
<td>SNMT</td>
<td>target – CCG</td>
<td>interleaving</td>
<td>32.0</td>
<td>32.7*</td>
<td>29.2</td>
<td>29.3**</td>
</tr>
<tr>
<td>Multitasking</td>
<td>target – CCG</td>
<td>shared encoder</td>
<td>31.4</td>
<td>32.0</td>
<td>28.4</td>
<td>29.0*</td>
</tr>
<tr>
<td>SNMT</td>
<td>source – dep</td>
<td>shared embedding</td>
<td>31.4</td>
<td>32.2</td>
<td>28.2</td>
<td>28.9</td>
</tr>
<tr>
<td></td>
<td>+ target – CCG</td>
<td>+ interleaving</td>
<td>32.1</td>
<td>33.0**</td>
<td>29.1</td>
<td>29.6**</td>
</tr>
</tbody>
</table>

Table 3: Experiments with target-side syntax for German→English and Romanian→English. BLEU scores reported for baseline NMT, syntax-aware NMT (SNMT) and multitasking. The SNMT system is also combined with source dependencies. Statistical significance is indicated with * $p < 0.05$ and ** $p < 0.01$, when comparing against the NMT baseline.

Initially, we had intended to use global syntax on the source-side as well for German→English, however the German CCG tree-bank is still under development.

From the results in Table 3 we observe that for German→English the source-side dependency labels improve BLEU by only 0.1, while Romanian→English sees an improvement of 0.5. Source-syntax may help more for Romanian→English because the training data is smaller and the word order is more similar between the source and target languages than it is for German→English.

For both language pairs, target-syntax improves translation quality more than source-syntax. However, target-syntax is complemented by source-syntax when used together, leading to a final improvement of 0.9 BLEU points for German→English and 1.2 BLEU points for Romanian→English.

Finally, we show that CCG supertags are also an effective representation of global-syntax when used in the encoder. In Table 4 we present results for using CCG supertags as source-syntax in the embedding layer. Because we have CCG annotations only for English, we reverse the translation directions and report BLEU scores for English→German and English→Romanian. The BLEU scores reported are for the ensemble models over newstest2016.

For English→German BLEU increases by 0.7 points and for English→Romanian by 0.5 points. In contrast, Sennrich and Haddow (2016) obtain an improvement of only 0.2 for English→German using dependency labels which encode only the grammatical function of words. These results confirm that representing global syntax in the encoder provides complementary information that the baseline NMT model is not able to learn from the source word sequence alone.

4.3 Analyses by sentence type

In this section, we make a finer grained analysis of the impact of target-side syntax by looking at a breakdown of BLEU scores with respect to different linguistic constructions and sentence lengths7.

We classify sentences into different linguistic constructions based on the CCG supertags that appear in them, e.g., the presence of category ⟨NP\NP⟩/(S/NP) indicates a subordinate construction. Figure 3 a) shows the difference in BLEU points between the syntax-aware NMT system and the baseline NMT system for the following linguistic constructions: coordination (conj), control and raising (control), prepositional phrase attachment (pp), questions and subordinate clauses (subordinate). In the figure we use the symbol "*" to indicate that syntactic information is used on the target (eg. de-en*), or both on the source and target (eg. *de-en*). We report the number of sentences for each category in Table 5.

With target-syntax, we see consistent improvements across all linguistic constructions for Romanian→English and across all but control and raising for German→English. In particular, the in-
Figure 3: Difference in BLEU points between SNMT and NMT, relative to baseline NMT scores, with respect to a) linguistic constructs and b) sentence lengths. The numbers attached to the bars represent the BLEU score for the baseline NMT system. The symbol * indicates that syntactic information is used on the target (eg. de-en*), or both on the source and target (eg. *de-en*).

Table 5: Sentence counts for different linguistic constructions.

<table>
<thead>
<tr>
<th></th>
<th>sub.</th>
<th>qu.</th>
<th>pp</th>
<th>contr.</th>
<th>conj</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO↔EN</td>
<td>742</td>
<td>90</td>
<td>1,572</td>
<td>415</td>
<td>845</td>
</tr>
<tr>
<td>DE↔EN</td>
<td>936</td>
<td>114</td>
<td>2,321</td>
<td>546</td>
<td>1,129</td>
</tr>
</tbody>
</table>

Table 5: Sentence counts for different linguistic constructions.

Table 6: Sentence counts for different sentence lengths.

<table>
<thead>
<tr>
<th></th>
<th>&lt;15</th>
<th>15-25</th>
<th>25-35</th>
<th>&gt;35</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO↔EN</td>
<td>491</td>
<td>540</td>
<td>433</td>
<td>520</td>
</tr>
<tr>
<td>DE↔EN</td>
<td>918</td>
<td>934</td>
<td>582</td>
<td>560</td>
</tr>
</tbody>
</table>
Our experiments demonstrate that target-syntax improves translation for two translation directions: German→English and Romanian→English. Our proposed method predicts the target words together with their CCG supertags.

Although the focus of this paper is not improving CCG tagging, we can also measure that SNMT is accurate at predicting CCG supertags. We compare the CCG sequence predicted by the SNMT models with that predicted by EasySRL and obtain the following accuracies: 93.2 for Romanian→English, 95.6 for German→English, 95.8 for German→English with both source and target syntax.8

We conclude by giving a couple of examples in Figure 4 for which the SNMT system with target syntax produced more grammatical translations than the baseline NMT system.

In the example DE-EN Question the baseline NMT system translates the preposition “über” twice as “about”. The SNMT system with target syntax predicts the correct CCG supertag for “what” which expects to be followed by a sentence and not a preposition: NP/([S[dcl]/NP]). Therefore the SNMT correctly re-orders the preposition “about” at the end of the question.

In the example DE-EN Subordinate the baseline NMT system fails to correctly attach “Prentiss” as an object and “his wife” as a modifier to the verb “called (bezeichnete)” in the subordinate clause. In contrast the SNMT system predicts the correct sub-categorization frame of the verb “described” and correctly translates the entire predicate-argument structure.

5 Conclusions

This work introduces a method for modeling explicit target-syntax in a neural machine translation system, by interleaving target words with their corresponding CCG supertags. Earlier work on syntax-aware NMT mainly modeled syntax in the encoder, while our experiments suggest modeling syntax in the decoder is also useful. Our results show that a tight integration of syntax in the decoder improves translation quality for both

---

8The multitasking model predicts a different number of CCG supertags than the number of target words. For the sentences where these numbers match, the CCG supertagging accuracy is 73.2.
German→English and Romanian→English language pairs, more so than a loose coupling of target words and syntax as in multitask learning. Finally, by combining our method for integrating target-syntax with the framework of Sennrich and Haddow (2016) for source-syntax we obtain the most improvement over the baseline NMT system: 0.9 BLEU for German→English and 1.2 BLEU for Romanian→English. In particular, we see large improvements for longer sentences involving syntactic phenomena such as subordinate and coordinate clauses and prepositional phrase attachment. In future work, we plan to evaluate the impact of target-syntax when translating into a morphologically rich language, for example by using the Hindi CCGBank (Ambati et al., 2016).

Acknowledgements

We thank the anonymous reviewers for their comments and suggestions. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements 644402 (HimL), 644333 (SUMMA) and 645452 (QT21).

References


Maria Nadejde, Alexandra Birch, and Philipp Koehn.


Rico Sennrich, Martin Volk, and Gerold Schneider. 2013. Exploiting Synergies Between Open Re-


