

Inducing Grammars with and for Neural Machine Translation

Ke Tran

Informatics Institute
University of Amsterdam
ketranmanh@gmail.com

Yonatan Bisk

Department of Computer Science
University of Washington
ybisk@cs.washington.com

Abstract

Machine translation systems require semantic knowledge and grammatical understanding. Neural machine translation (NMT) systems often assume this information is captured by an attention mechanism and a decoder that ensures fluency. Recent work has shown that incorporating explicit syntax alleviates the burden of modeling both types of knowledge. However, requiring parses is expensive and does not explore the question of what syntax a model needs during translation. To address both of these issues we introduce a model that simultaneously translates while inducing dependency trees. In this way, we leverage the benefits of structure while investigating what syntax NMT must induce to maximize performance. We show that our dependency trees are 1. language pair dependent and 2. improve translation quality.

1 Motivation

Language has syntactic structure and translation models need to understand grammatical dependencies to resolve the semantics of a sentence and preserve agreement (e.g., number, gender, etc). Many current approaches to MT have been able to avoid explicitly providing structural information by relying on advances in sequence to sequence (seq2seq) models. The most famous advances include attention mechanisms (Bahdanau et al., 2015) and gating in Long Short-Term Memory (LSTM) cells (Hochreiter and Schmidhuber, 1997).

In this work we aim to benefit from syntactic structure, without providing it to the model, and to disentangle the semantic and syntactic components of translation, by introducing a gating mechanism which controls when syntax should be used.

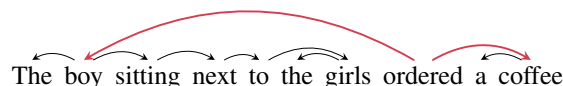


Figure 1: Our model aims to capture both:
syntactic (verb *ordered* → subj/obj *boy, coffee*)
alignment (noun *girls* → determiner *the*) attention.

Consider the process of translating the sentence “*The boy sitting next to the girls ordered a coffee.*” (Figure 1) from English to German. In German, translating *ordered*, requires knowledge of its subject *boy* to correctly predict the verb’s number *bestellte* instead of *bestellten*. This is a case where syntactic agreement requires long-distance information. On the other hand, *next* can be translated in isolation. The model should uncover these relationships and decide when and which aspects of syntax are necessary. While in principle decoders can utilize previously predicted words (e.g., the translation of *boy*) to reason about subject-verb agreement, in practice LSTMs still struggle with long-distance dependencies. Moreover, Belinkov et al. (2017) showed that using attention reduces the decoder’s capacity to learn target side syntax.

In addition to demonstrating improvements in translation quality, we are also interested in analyzing the predicted dependency trees discovered by our models. Recent work has begun analyzing task-specific latent trees (Williams et al., 2018). We present the first results on learning latent trees with a joint syntactic-semantic objective. We do this in the service of machine translation which inherently requires access to both aspects of a sentence. Further, our results indicate that language pairs with rich morphology require and therefore induce more complex syntactic structure.

Our use of a structured self attention encoder (§4) that predicts a non-projective dependency tree

over the source sentence provides a soft structured representation of the source sentence that can then be transferred to the decoder, which alleviates the burden of capturing target syntax on the target side.

We will show that the quality of the induced trees depends on the choice of the target language (§7). Moreover, the gating mechanism will allow us to examine which contexts require source side syntax.

In summary, in this work:

- We propose a new NMT model that discovers latent structures for encoding and when to use them, while achieving significant improvements in BLEU scores over a strong baseline.
- We perform an in-depth analysis of the induced structures and investigate where the target decoder decides syntax is required.

2 Related Work

Recent work has begun investigating what syntax seq2seq models capture (Linzen et al., 2016), but this is evaluated via downstream tasks designed to test the model’s abilities and not its representation.

Simultaneously, recent research in neural machine translation (NMT) has shown the benefit of modeling syntax explicitly (Aharoni and Goldberg, 2017; Bastings et al., 2017; Li et al., 2017; Eriguchi et al., 2017) rather than assuming the model will automatically discover and encode it.

Bradbury and Socher (2017) presented an encoder-decoder architecture based on RNNG (Dyer et al., 2016). However, their preliminary work was not scaled to a large MT dataset and omits analysis of the induced trees.

Unlike the previous work on source side latent graph parsing (Hashimoto and Tsuruoka, 2017), our structured self attention encoder allows us to extract a dependency tree in a principled manner. Therefore, learning the internal representation of our model is related to work done in unsupervised grammar induction (Klein and Manning, 2004; Spitkovsky et al., 2011) except that by focusing on translation we require both syntactic and semantic knowledge.

In this work, we attempt to contribute to both modeling syntax and investigating a more interpretable interface for testing the syntactic content of a new seq2seq models’ internal representation.

3 Neural Machine Translation

Given a training pair of source and target sentences (\mathbf{x}, \mathbf{y}) of length n and m respectively, neural machine translation is a conditional probabilistic model $p(\mathbf{y} | \mathbf{x})$ implemented using neural networks

$$\log p(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}) = \sum_{j=1}^m \log p(y_j | \mathbf{y}_{i < j}, \mathbf{x}; \boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ is the model’s parameters. We will omit the parameters $\boldsymbol{\theta}$ herein for readability.

The NMT system used in this work is a seq2seq model that consists of a bidirectional LSTM encoder and an LSTM decoder coupled with an attention mechanism (Bahdanau et al., 2015; Luong et al., 2015). Our system is based on a PyTorch implementation¹ of OpenNMT (Klein et al., 2017). Let $\{\mathbf{s}_i \in \mathbb{R}^d\}_{i=1}^n$ be the output of the encoder

$$\mathbf{S} = \text{BiLSTM}(\mathbf{x}) \quad (1)$$

Here we use $\mathbf{S} = [\mathbf{s}_1; \dots; \mathbf{s}_n] \in \mathbb{R}^{d \times n}$ as a concatenation of $\{\mathbf{s}_i\}$. The decoder is composed of stacked LSTMs with input-feeding. Specifically, the inputs of the decoder at time step t are a concatenation of the embedding of the previous generated word \mathbf{y}_{t-1} and a vector \mathbf{u}_{t-1} :

$$\mathbf{u}_{t-1} = g(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \quad (2)$$

where g is a one layer feed-forward network, \mathbf{h}_{t-1} is the output of the LSTM decoder, and \mathbf{c}_{t-1} is a context vector computed by an attention mechanism

$$\boldsymbol{\alpha}_{t-1} = \text{softmax}(\mathbf{h}_{t-1}^T \mathbf{W}_a \mathbf{S}) \quad (3)$$

$$\mathbf{c}_{t-1} = \mathbf{S} \boldsymbol{\alpha}_{t-1}^T \quad (4)$$

where $\mathbf{W}_a \in \mathbb{R}^{d \times d}$ is a trainable parameter.

Finally a single layer feed-forward network f takes \mathbf{u}_t as input and returns a multinomial distribution over all the target words: $y_t \sim f(\mathbf{u}_t)$

4 Syntactic Attention Model

We propose a syntactic attention model² (Figure 2) that differs from standard NMT in two crucial aspects. First, our encoder outputs two sets of annotations: content annotations \mathbf{S} and syntactic annotations \mathbf{M} (Figure 2a). The content annotations are the outputs of a standard BiLSTM while the

¹<http://opennmt.net/OpenNMT-py/>

²<https://github.com/ketranm/sa-nmt>

syntactic annotations are produced by a head word selection layer (§4.1). The syntactic annotations \mathbf{M} capture syntactic dependencies amongst the source words and enable syntactic transfer from the source to the target. Second, we incorporate the source side syntax into our model by modifying the standard attention (from target to source) in NMT such that it attends to both \mathbf{S} and \mathbf{M} through a *shared attention* layer. The shared attention layer biases our model toward capturing source side dependency. It produces a dependency context \mathbf{d} (Figure 2c) in addition to the standard context vector \mathbf{c} (Figure 2b) at each time step. Motivated by the example in Figure 1 that some words can be translated without resolving their syntactic roles in the source sentence, we include a gating mechanism that allows the decoder to decide the amount of syntax needed when it generates the next word. Next, we describe the head word selection layer and how source side syntax is incorporated into our model.

4.1 Head Word Selection

The head word selection layer learns to select a *soft* head word for each source word. This layer transforms \mathbf{S} into a matrix \mathbf{M} that encodes implicit dependency structure of \mathbf{x} using *structured self attention*. First we apply three trainable weight matrices $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d}$ to map \mathbf{S} to query, key, and value matrices $\mathbf{S}_q = \mathbf{W}_q \mathbf{S}, \mathbf{S}_k = \mathbf{W}_k \mathbf{S}, \mathbf{S}_v = \mathbf{W}_v \mathbf{S} \in \mathbb{R}^{d \times n}$ respectively. Then we compute the structured self attention probabilities $\beta \in \mathbb{R}^{n \times n}$ via a function *sattn*: $\beta = \text{sattn}(\mathbf{S}_q^T \mathbf{S}_k / \sqrt{d})$. Finally the syntactic context \mathbf{M} is computed as $\mathbf{M} = \mathbf{S}_v \beta$.

Here n is the length of the source sentence, so β captures all pairwise word dependencies. Each cell $\beta_{i,j}$ of the attention matrix β is the posterior probability $p(x_i = \text{head}(x_j) | \mathbf{x})$. The structured self attention function *sattn* is inspired by the work of (Kim et al., 2017) but differs in two important ways. First we model *non-projective dependency trees*. Second, we utilize the Kirchhoff’s Matrix-Tree Theorem (Tutte, 1984) instead of the sum-product algorithm presented in (Kim et al., 2017) for fast evaluation of the attention probabilities. We note that (Liu and Lapata, 2018) were first to propose using the Matrix-Tree Theorem for evaluating the marginals in end to end training of neural networks. Their work, however, focuses on the task of natural language inference (Bowman et al., 2015) and document classification which arguably require less syntactic knowledge than machine translation.

Additionally, we will evaluate our structured self attention on datasets that are up to 20 times larger than the datasets studied in previous work.

Let $\mathbf{z} \in \{0, 1\}^{n \times n}$ be an adjacency matrix encoding a source’s dependency tree. Let $\phi = \mathbf{S}_q^T \mathbf{S}_k / \sqrt{d} \in \mathbb{R}^{n \times n}$ be a scoring matrix such that cell $\phi_{i,j}$ scores how likely word x_i is to be the head of word x_j . The probability of a dependency tree \mathbf{z} is therefore given by

$$p(\mathbf{z} | \mathbf{x}; \phi) = \frac{\exp\left(\sum_{i,j} \mathbf{z}_{i,j} \phi_{i,j}\right)}{Z(\phi)} \quad (5)$$

where $Z(\phi)$ is the partition function.

In the head selection model, we are interested in the marginal $p(\mathbf{z}_{i,j} = 1 | \mathbf{x}; \phi)$

$$\beta_{i,j} = p(\mathbf{z}_{i,j} = 1 | \mathbf{x}; \phi) = \sum_{\mathbf{z} : \mathbf{z}_{i,j}=1} p(\mathbf{z} | \mathbf{x}; \phi)$$

We use the framework presented by Koo et al. (2007) to compute the marginal of non-projective dependency structures. Koo et al. (2007) use the Kirchhoff’s Matrix-Tree Theorem (Tutte, 1984) to compute $p(\mathbf{z}_{i,j} = 1 | \mathbf{x}; \phi)$ by first defining the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ as follows:

$$\mathbf{L}_{i,j}(\phi) = \begin{cases} \sum_{\substack{k=1 \\ k \neq j}}^n \exp(\phi_{k,j}) & \text{if } i = j \\ -\exp(\phi_{i,j}) & \text{otherwise} \end{cases} \quad (6)$$

Now we construct a matrix $\hat{\mathbf{L}}$ that accounts for root selection

$$\hat{\mathbf{L}}_{i,j}(\phi) = \begin{cases} \exp(\phi_{j,j}) & \text{if } i = 1 \\ \mathbf{L}_{i,j}(\phi) & \text{if } i > 1 \end{cases} \quad (7)$$

The marginals in β are then

$$\beta_{i,j} = (1 - \delta_{1,j}) \exp(\phi_{i,j}) \left[\hat{\mathbf{L}}^{-1}(\phi) \right]_{j,j} - (1 - \delta_{i,1}) \exp(\phi_{i,j}) \left[\hat{\mathbf{L}}^{-1}(\phi) \right]_{j,i} \quad (8)$$

where $\delta_{i,j}$ is the Kronecker delta. For the root node, the marginals are given by

$$\beta_{k,k} = \exp(\phi_{k,k}) \left[\hat{\mathbf{L}}^{-1}(\phi) \right]_{k,1} \quad (9)$$

The computation of the marginals is fully differentiable, thus we can train the model in an end-to-end fashion by maximizing the conditional likelihood of the translation.

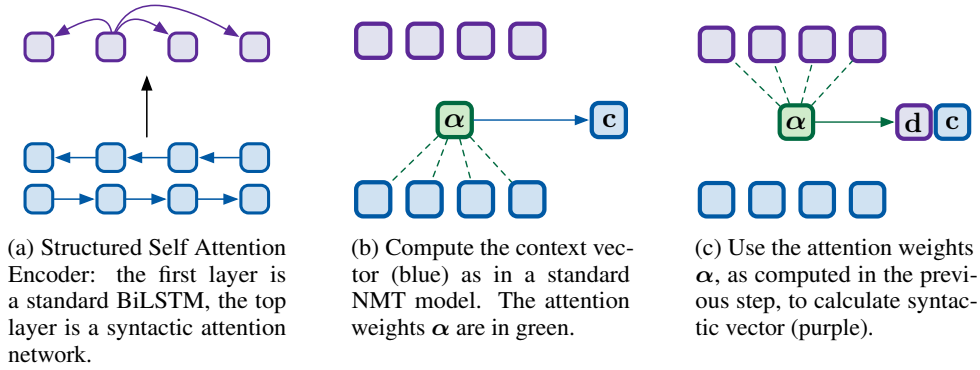


Figure 2: A visual representation of our proposed mechanism for shared attention.

4.2 Incorporating Syntactic Context

Having set the annotations \mathbf{S} and \mathbf{M} with the encoder, the LSTM decoder can utilize this information at every generation step by means of attention. At time step t , we first compute standard attention weights α_{t-1} and context vector \mathbf{c}_{t-1} as in Equations (3) and (4). We then compute a weighted syntactic vector:

$$\mathbf{d}_{t-1} = \mathbf{M}\alpha_{t-1}^\top \quad (10)$$

Note that the syntactic vector \mathbf{d}_{t-1} and the context vector \mathbf{c}_{t-1} share the same attention weights α_{t-1} . The main idea behind sharing attention weights (Figure 2c) is that if the model attends to a particular source word x_i when generating the next target word, we also want the model to attend to the head word of x_i . We share the attention weights α_{t-1} because we expect that, if the model picks a source word x_i to translate with the highest probability $\alpha_{t-1}[i]$, the contribution of x_i 's head in the syntactic vector \mathbf{d}_{t-1} should also be highest. Figure 3

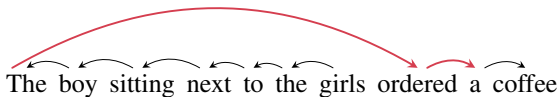


Figure 3: A latent tree learned by our model.

shows the latent tree learned by our translation objective. Unlike the gold tree provided in Figure 1, the model decided that “*the boy*” is the head of “*ordered*”. This is common in our model because the BiLSTM context means that a given word’s representation is actually a summary of its local context/constituent.

It is not always useful or necessary to access the syntactic context \mathbf{d}_{t-1} at every time step t . Ideally,

we should let the model decide whether it needs to use this information or not. For example, the model might decide to only use syntax when it needs to resolve long distance dependencies on the source side. To control the amount of source side syntactic information, we introduce a gating mechanism:

$$\hat{\mathbf{d}}_{t-1} = \mathbf{d}_{t-1} \odot \sigma(\mathbf{W}_g \mathbf{h}_{t-1}) \quad (11)$$

The vector \mathbf{u}_{t-1} from Eq. (2) now becomes

$$\mathbf{u}_{t-1} = g(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}, \hat{\mathbf{d}}_{t-1}) \quad (12)$$

Another approach to incorporating syntactic annotations \mathbf{M} in the decoder is to use a separate attention layer to compute the syntactic vector \mathbf{d}_{t-1} at time step t :

$$\gamma_{t-1} = \text{softmax}(\mathbf{h}_{t-1}^\top \mathbf{W}_m \mathbf{M}) \quad (13)$$

$$\mathbf{d}_{t-1} = \mathbf{M}\gamma_{t-1}^\top \quad (14)$$

We will provide a comparison to this approach in our results.

4.3 Hard Attention over Tree Structures

Finally, to simulate the scenario where the model has access to a dependency tree given by an external parser we report results with hard attention. Forcing the model to make hard decisions during training mirrors the extraction and conditioning on a dependency tree (§7.1). We expect this technique will improve the performance on grammar induction, despite making translation lossy. A similar observation has been reported in (Hashimoto and Tsuruoka, 2017) which showed that translation performance degraded below their baseline when they provided dependency trees to the encoder.

Recall the marginal $\beta_{i,j}$ gives us the probability that word x_i is the head of word x_j . We convert

these soft weights to hard ones $\bar{\beta}$ by

$$\bar{\beta}_{k,j} = \begin{cases} 1 & \text{if } k = \arg \max_i \beta_{i,j} \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

We train this model using the straight-through estimator (Bengio et al., 2013). In this setup, each word has a parent but there is no guarantee that the structure given by hard attention will result in a tree (*i.e.*, it may contain cycle). A more principled way to enforce a tree structure is to decode the best tree \mathcal{T} using the maximum spanning tree algorithm (Chu and Liu, 1965; Edmonds, 1967) and to set $\bar{\beta}_{k,j} = 1$ if the edge $(x_k \rightarrow x_j) \in \mathcal{T}$. Maximum spanning tree decoding can be prohibitively slow as the Chu-Liu-Edmonds algorithm is not GPU friendly. We therefore greedily pick a parent word for each word x_j in the sentence using Eq. (15). This is actually a principled simplification as greedily assigning a parent for each word is the first step in Chu-Liu-Edmonds algorithm.

5 Experiments

Next we will discuss our experimental setup and report results for English \leftrightarrow German (En \leftrightarrow De), English \leftrightarrow Russian (En \leftrightarrow Ru), and Russian \rightarrow Arabic (Ru \rightarrow Ar) translation models.

5.1 Data

We use the WMT17 (Bojar et al., 2017) data in our experiments. Table 1 shows the statistics of the data. For En \leftrightarrow De, we use a concatenation of Europarl, Common Crawl, Rapid corpus of EU press releases, and News Commentary v12. We use *newstest2015* for development and *newstest2016*, *newstest2017* for testing. For En \leftrightarrow Ru, we use Common Crawl, News Commentary v12, and Yandex Corpus. The development data comes from *newstest2016* and *newstest2017* is reserved for testing. For Ru \rightarrow Ar, we use the data from the six-way sentence-aligned subcorpus of the United Nations Parallel Corpus v1.0 (Ziems et al., 2016). The corpus also contains the official development and test data. Our lan-

	Train	Valid	Test	Vocabulary
En \leftrightarrow De	5.9M	2,169	2,999 / 3,004	36,251 / 35,913
En \leftrightarrow Ru	2.1M	2,998	3,001	34,872 / 34,989
Ru \rightarrow Ar	11.1M	4,000	4,000	32,735 / 32,955

Table 1: Statistics of the data.

guage pairs were chosen to compare results across

and between morphologically rich and poor languages. This will prove particularly interesting in our grammar induction results where different pairs must preserve different amounts of syntactic agreement information.

We use BPE (Sennrich et al., 2016) with 32,000 merge operations. We run BPE for each language instead of using BPE for the concatenation of both source and target languages.

5.2 Baselines

Our baseline is an NMT model with input-feeding (§3). As we will be making several modifications from the basic architecture in our proposed structured self attention NMT (SA-NMT), we will verify each choice in our architecture design empirically. First we validate the structured self attention module by comparing it to a self-attention module (Lin et al., 2017; Vaswani et al., 2017). Self attention computes attention weights β simply as $\beta = \text{softmax}(\phi)$. Since self-attention does not assume any hierarchical structure over the source sentence, we refer it as flat-attention NMT (FA-NMT). Second, we validate the benefit of using two sets of annotations in the encoder. We combine the hidden states of the encoder \mathbf{h} with syntactic context \mathbf{d} to obtain a single set of annotation using the following equation:

$$\bar{\mathbf{s}}_i = \mathbf{s}_i + \sigma(\mathbf{W}_g \mathbf{s}_i) \odot \mathbf{d}_i \quad (16)$$

Here we first down-weight the syntactic context \mathbf{d}_i before adding it to \mathbf{s}_i . The sigmoid function $\sigma(\mathbf{W}_g \mathbf{s}_i)$ decides the weight of the head word of x_i based on whether translating x_i needs additionally dependency information. We refer to this baseline as SA-NMT-1set. Note that in this baseline, there is only one attention layer from the target to the source $\bar{\mathbf{S}} = \{\bar{\mathbf{s}}_i\}_1^n$.

In all the models, we share the weights of target word embeddings and the output layer as suggested by Inan et al. (2017) and Press and Wolf (2017).

5.3 Hyper-parameters and Training

For all the models, we set the word embedding size to 1024, the number of LSTM layers to 2, and the dropout rate to 0.3. Parameters are initialized uniformly in $(-0.04, 0.04)$. We use the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 0.001. We evaluate our models on development data every 10,000 updates for De-En and Ru \rightarrow Ar, and 5,000 updates for Ru-En. If the

validation perplexity increases, we decay the learning rate by 0.5. We stop training after decaying the learning rate five times as suggested by Denkowski and Neubig (2017). The mini-batch size is 64 in Ru→Ar experiments and 32 in the rest. Finally, we report BLEU scores computed using the standard `multi-bleu.perl` script.

In our experiments, the SA-NMT models are twice slower than the baseline NMT measuring by the number of target words generated per second.

5.4 Translation Results

Table 2 shows the BLEU scores in our experiments. We test statistical significance using bootstrap resampling (Riezler and Maxwell, 2005). Statistical significances are marked as $\dagger p < 0.05$ and $\ddagger p < 0.01$ when compared against the baselines. Additionally, we also report statistical significances $\Delta p < 0.05$ and $\blacktriangle p < 0.01$ when comparing against the FA-NMT models that have two separate attention layers from the decoder to the encoder. Overall, the SA-NMT (shared) model performs the best gaining more than 0.5 BLEU De→En on *wmt16*, up to 0.82 BLEU on En→De *wmt17* and 0.64 BLEU En→Ru direction over a competitive NMT baseline. The gain of the SA-NMT model on Ru→Ar is small (0.45 BLEU) but significant. The results show that structured self attention is useful when translating from English to languages that have long-distance dependencies and complex morphological agreements. We also see that the gain is marginal compared to self-attention models (FA-NMT-shared) and not significant. Within FA-NMT models, sharing attention is helpful. Our results also confirm the advantage of having two separate sets of annotations in the encoder when modeling syntax. The hard structured self attention model (SA-NMT-hard) performs comparably to the baseline. While this is a somewhat expected result from the hard attention model, we will show in Section 7 that the quality of induced trees from hard attention is often far better than those from soft attention.

6 Gate Activation Visualization

As mentioned earlier, our models allow us to ask the question: When does the target LSTM need to access source side syntax? We investigate this by analyzing the gate activations of our best model, SA-NMT (shared). At time step t , when the model is about to predict the target word y_t , we compute

the norm of the gate activations

$$z_t = \|\sigma(\mathbf{W}_g \mathbf{h}_{t-1})\|_2 \quad (17)$$

The activation norm z_t allows us to see how much syntactic information flows into the decoder. We observe that z_t has its highest value when the decoder is about to generate a verb while it has its lowest value when the end of sentence token `</s>` is predicted. Figure 4 shows some examples of German target sentences. The darker colors represent higher activation norms.

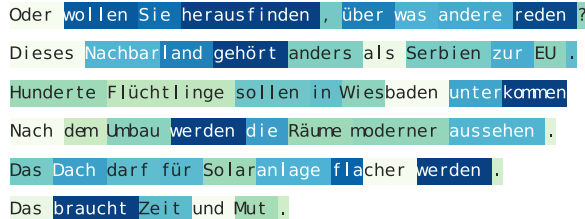
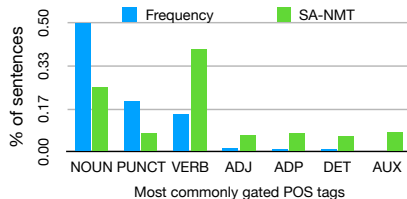


Figure 4: Visualization of gate norm. Darker means the model is using more syntactic information.

It is clear that translating verbs requires structural information. We also see that after verbs, the gate activation norms are highest at nouns *Zeit* (*time*), *Mut* (*courage*), *Dach* (*roof*) and then tail off as we move to function words which require less context to disambiguate. Below are the frequencies with which the highest activation norm in a sentence is applied to a given part-of-speech tag on *newstest2016*. We include the top 7 most common activations. We see that while nouns are often the most common tag in a sentence, syntax is disproportionately used for translating verbs.



7 Grammar Induction

NLP has long assumed hierarchical structured representations are important to understanding language. In this work, we borrowed that intuition to inform the construction of our model. We investigate whether the internal latent representations discovered by our models share properties previously identified within linguistics and if not, what important differences exist. We investigate the interpretability of our model’s representations by: 1)

Model	Shared	De→En		Ru→En	En→De		En→Ru	Ru→Ar
		wmt16	wmt17	wmt17	wmt16	wmt17	wmt17	un-test
NMT	-	33.16	28.94	30.17	29.92	23.44	26.41	37.04
FA-NMT	yes	33.55	29.43	30.22	30.09	24.03	26.91	37.41
	no	33.24	29.00	30.34	29.98	23.97	26.75	37.20
SA-NMT-1set	-	33.51	29.15	30.34	30.29 [†]	24.12	26.96	37.34
SA-NMT-hard	yes	33.38	28.96	29.98	29.93	23.84	26.71	37.33
SA-NMT	yes	33.73 ^{‡Δ}	29.45 ^{‡▲}	30.41	30.22	24.26 ^{‡Δ}	27.05 [‡]	37.49 ^{‡Δ}
	no	33.18	29.19	30.15	30.17	23.94	27.01	37.22

Table 2: Results for translating En↔De, En↔Ru, and Ru→Ar. Statistical significances are marked as [†] $p < 0.05$ and [‡] $p < 0.01$ when compared against the baselines and ^Δ/[▲] when compared against the FA-NMT (no-shared). The results indicate the strength of our proposed *shared-attention* for NMT.

	FA		SA			Baseline		
	no-shared	shared	no-shared	shared	hard	L	R	Un
EN (→DE)	17.0/25.2	27.6/41.3	23.6/33.7	27.8/42.6	31.7/45.6	34.0	7.8	40.9
EN (→RU)	35.2/48.5	36.5/48.8	12.8/25.5	33.1/48.9	33.7/46.0			
DE (→EN)	21.1/33.3	20.1/33.6	12.8/22.5	21.5/38.0	26.3/40.7	34.4	8.6	41.5
RU (→EN)	19.2/33.2	20.4/34.9	19.3/34.4	24.8/41.9	23.2/33.3			
RU (→AR)	21.1/41.1	22.2/42.1	11.6/21.4	28.9/50.4	30.3/52.0	32.9	15.2	47.3

Table 3: Directed and Undirected (DA/UA) model accuracy (without punctuation) compared to branching baselines: left (L), right (R) and undirected (Un). Our results show an intriguing effect of the target language on induction. Note the accuracy discrepancy between translating RU to EN versus AR.

A quantitative attachment accuracy and 2) A qualitative look at its output.

Our results corroborate and refute previous work (Hashimoto and Tsuruoka, 2017; Williams et al., 2018). We provide stronger evidence that syntactic information can be discovered via latent structured self attention, but we also present preliminary results indicating that conventional definitions of syntax may be at odds with task specific performance.

Unlike in the grammar induction literature our model is not specifically constructed to recover traditional dependency grammars nor have we provided the model with access to part-of-speech tags or universal rules (Naseem et al., 2010; Bisk and Hockenmaier, 2013). The model only uncovers the syntactic information necessary for a given language pair, though future work should investigate if structural linguistic constraints benefit MT.

7.1 Extracting a Tree

For extracting non-projective dependency trees, we use Chu-Liu-Edmonds algorithm (Chu and Liu, 1965; Edmonds, 1967). First, we must collapse BPE segments into words. Assume the k -th word corresponds to BPE tokens from index u to v . We obtain a new matrix $\hat{\phi}$ by summing over $\phi_{i,j}$ that

are the corresponding BPE segments.

$$\hat{\phi}_{i,j} = \begin{cases} \phi_{i,j} & \text{if } i \notin [u, v] \wedge j \notin [u, v] \\ \sum_{l=u}^v \phi_{i,l} & \text{if } j = k \wedge i \notin [u, v] \\ \sum_{l=u}^v \phi_{l,j} & \text{if } i = k \wedge j \notin [u, v] \\ \sum_{l,h=u}^v \phi_{l,h} & \text{otherwise} \end{cases}$$

7.2 Grammatical Analysis

To analyze performance we compute unlabeled directed and undirected attachment accuracies of our predicted trees on gold annotations from the Universal Dependencies (UD version 2) dataset.³ We chose this representation because of its availability in many languages, though it is atypical for grammar induction. Our five model settings in addition to left and right branching baselines are presented in Table 3. The results indicate that the target language effects the source encoder’s induction performance and several settings are competitive with branching baselines for determining headedness. Recall that syntax is being modeled on the source language so adjacent rows are comparable.

We observe a huge boost in DA/UA scores for EN and RU in FA-NMT and SA-NMT-shared models when the target languages are morphologically

³<http://universaldependencies.org>

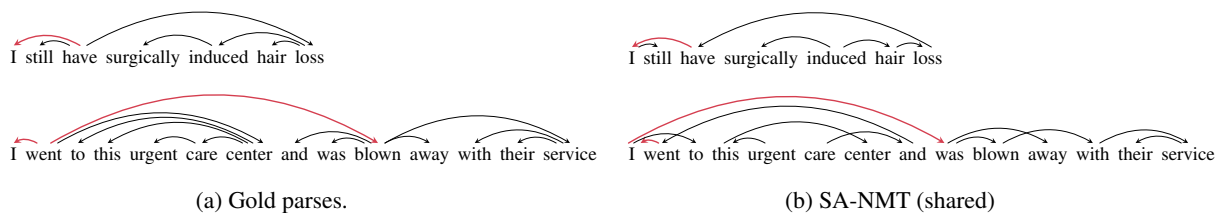


Figure 6: Samples of induced trees for English by our (En→Ru) model. Notice the red arrows from *subject*↔*verb* which are necessary for translating Russian verbs.

rich (RU and AR respectively). In comparison to previous work (Belinkov et al., 2017; Shi et al., 2016) on an encoder’s ability to capture source side syntax, we show a stronger result that even when the encoders are designed to capture syntax explicitly, the choice of the target language influences the amount of syntax learned by the encoder.

We also see gains from hard attention and several models outperform baselines for undirected dependency metrics (UA). Whether hard attention helps in general is unclear. It appears to help when the target languages are morphologically rich.

Successfully extracting linguistic structure with hard attention indicates that models can capture interesting structures beyond semantic co-occurrence via discrete actions. Our approach also outperforms (Hashimoto and Tsuruoka, 2017) despite lacking access to additional resources like POS tags.⁴

7.3 Dependency Accuracies & Discrepancies

While the SA-NMT-hard model gives the best directed attachment scores on EN→DE, DE→EN and RU→AR, the BLEU scores of this model are below other SA-NMT models as shown in Table 2. The lack of correlation between syntactic performance and NMT contradicts the intuition of previous work and suggests that useful structures learned in service of a task might not necessarily benefit from or correspond directly to known linguistic formalisms. We want to raise three important differences between these induced structures and UD.

First, we see a blurred boundary between dependency and constituency representations. As noted earlier, the BiLSTM provides a local summary. When the model chooses a head word, it is actually choosing hidden states from a BiLSTM and therefore gaining access to a constituent or region. This means there is likely little difference between attending to the noun vs the determiner in

⁴The numbers are not directly comparable since they use WSJ corpus to evaluate the UA score.

a phrase (despite being wrong according to UD). Future work might force this distinction by replacing the BiLSTM with a bag-of-words but this will likely lead to substantial losses in MT performance.

Second, because the model appears to use syntax for agreement, often verb dependencies link to subjects directly to capture predicate argument structures like those in CCG or semantic role labeling. UD instead follows the convention of attaching all verbs that share a subject to one another or their conjunctions. We have colored some subject–verb links in Figure 6: *e.g.*, between *I*, *went* and *was*.

Finally, the model’s notion of headedness is atypical as it roughly translates to “helpful when translating”. The head word gets incorporated into the shared representation which may cause the arrow to flip from traditional formalisms. Additionally, because the model can turn on and off syntax as necessary, it is likely to produce high confidence treelets rather than complete parses. This means arcs produced from words with weak gate activations (Figure 4) are not actually used during translation and likely not-syntactically meaningful.

We will not speculate if these are desirable properties or issues to address with constraints, but the model’s decisions appear well motivated and our formulation allows us to have the discussion.

8 Conclusion

We have proposed a structured self attention encoder for NMT. Our models show significant gains in performance over a strong baseline on standard WMT benchmarks. The models presented here do not access any external information such as parse-trees or part-of-speech tags yet appear to use and induce structure when given the opportunity. Finally, we see our induction performance is language pair dependent, which invites an interesting research discussion as to the role of syntax in translation and the importance of working with morphologically rich languages.

Acknowledgments

We thank Joachim Daiber, Ekaterina Garmash, and Julia Kiseleva for helping with German and Russian examples. We are grateful to Arianna Bisazza, Miloš Stanojević, and Raquel Garrido Alhama for providing feedback on the draft. The second author was supported by Samsung Research.

References

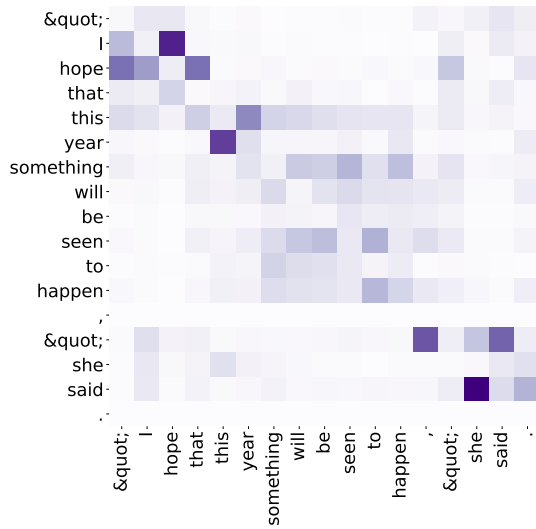
- Roei Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140, Vancouver, Canada. Association for Computational Linguistics.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*, San Diego, CA, USA.
- Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1947–1957. Association for Computational Linguistics.
- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872. Association for Computational Linguistics.
- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv e-prints*.
- Yonatan Bisk and Julia Hockenmaier. 2013. An HDP Model for Inducing Combinatory Categorical Grammars. *Transactions of the Association for Computational Linguistics*, pages 75–88.
- 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, Copenhagen, Denmark. Association for Computational Linguistics.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.
- James Bradbury and Richard Socher. 2017. Towards neural machine translation with latent tree attention. In *Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing*, pages 12–16, Copenhagen, Denmark. Association for Computational Linguistics.
- Y. J. Chu and T. H. Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14.
- Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27, Vancouver. Association for Computational Linguistics.
- Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. **Recurrent neural network grammars**. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.
- Jack Edmonds. 1967. Optimum Branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240.
- Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78. Association for Computational Linguistics.
- Kazuma Hashimoto and Yoshimasa Tsuruoka. 2017. Neural machine translation with source-side latent graph parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 125–135. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *ICLR*.
- Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. In *ICLR*.
- Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, San Diego, CA, USA.
- Dan Klein and Christopher D Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the*

- 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume, pages 478–485, Barcelona, Spain.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.
- Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *EMNLP*, pages 141–150, Prague, Czech Republic. 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 Association for Computational Linguistics (Volume 1: Long Papers)*, pages 688–697. Association for Computational Linguistics.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *ICLR*.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.
- Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244, Cambridge, MA.
- Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163. Association for Computational Linguistics.
- Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. 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 Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.
- Valentin I Spitkovsky, Hiyan Alshawi, Angel X Chang, and Daniel Jurafsky. 2011. Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1281–1290, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- W. T Tutte. 1984. *Graph theory*. Cambridge University Press.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.
- Adina Williams, Andrew Drozdov, and Samuel R. Bowman. 2018. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics*.
- Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

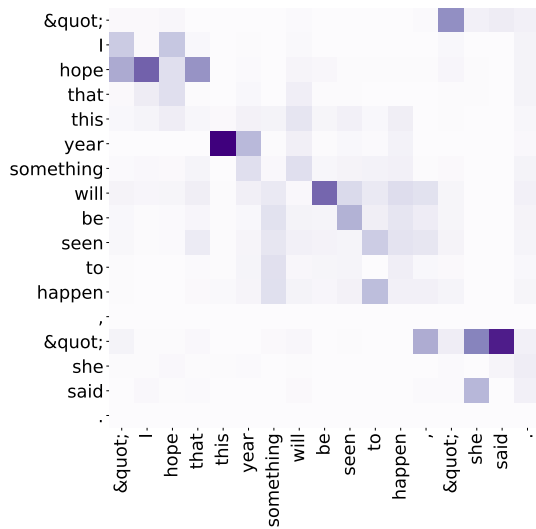
A Attention Visualization

Figure 7 shows a sample visualization of structured attention models trained on En→De data. It is worth noting that the shared SA-NMT model (Figure 7a) and the hard SA-NMT model (Figure 7b) capture similar structures of the source sentence. We hypothesize that when the objective function requires syntax, the induced trees are more consistent unlike those discovered by a semantic objective (Williams et al., 2018). Both models correctly identify that the verb is the head of pronoun (hope→I,

said→she). While intuitively it is clearly beneficial to know the subject of the verb when translating from English into German, the model attention is still somewhat surprising because long distance dependency phenomena are less common in English, so we would expect that a simple content based addressing (*i.e.*, standard attention mechanism) would be sufficient in this translation



(a) SA-NMT (shared) attention.



(b) SA-NMT with hard structured attention.

Figure 7: A visualization of attention distributions over head words (on y-axis).