Unsupervised Source Hierarchies for Low-Resource Neural Machine Translation

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

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

Abstract

Incorporating source syntactic information into neural machine translation (NMT) has recently proven successful (Eriguchi et al., 2016; Luong et al., 2016). However, this is generally done using an outside parser to syntactically annotate the training data, making this technique difficult to use for languages or domains for which a reliable parser is not available. In this paper, we introduce an unsupervised tree-to-sequence (tree2seq) model for neural machine translation; this model is able to induce an unsupervised hierarchical structure on the source sentence based on the downstream task of neural machine translation. We adapt the Gumbel tree-LSTM of Choi et al. (2018) to NMT in order to create the encoder.

We evaluate our model against sequential and supervised parsing baselines on three low- and medium-resource language pairs. For low-resource cases, the unsupervised tree2seq encoder significantly outperforms the baselines; no improvements are seen for medium-resource translation.

1 Introduction

Neural machine translation (NMT) is a widely used approach to machine translation that is often trained without outside linguistic information. In NMT, sentences are typically modeled using recurrent neural networks (RNNs), so they are represented in a continuous space, alleviating the sparsity issue that afflicted many previous machine translation approaches. As a result, NMT is state-of-the-art for many language pairs (Bentivogli et al., 2016; Toral and Sánchez-Cartagena, 2017).

Despite these successes, there is room for improvement. RNN-based NMT is sequential, whereas natural language is hierarchical; thus, RNNs may not be the most appropriate models for language. In fact, these sequential models do not fully learn syntax (Bentivogli et al., 2016; Linzen et al., 2016; Shi et al., 2016). In addition, although NMT performs well on high-resource languages, it is less successful in low-resource scenarios (Koehn and Knowles, 2017).

As a solution to these challenges, researchers have incorporated syntax into NMT, particularly on the source side. Notably, Eriguchi et al. (2016) introduced a tree-to-sequence (tree2seq) NMT model in which the RNN encoder was augmented with a tree long short-term memory (LSTM) network (Tai et al., 2015). This and related techniques have yielded improvements in NMT; however, injecting source syntax into NMT requires parsing the training data with an external parser, and such parsers may be unavailable for low-resource languages. Adding syntactic source information may improve low-resource NMT, but we would need a way of doing so without an external parser.

We would like to mimic the improvements that come from adding source syntactic hierarchies to NMT without requiring syntactic annotations of the training data. Recently, there have been some proposals to induce unsupervised hierarchies based on semantic objectives for sentiment analysis and natural language inference (Choi et al., 2018; Yogatama et al., 2017). Here, we apply these hierarchical sentence representations to low-resource neural machine translation.

In this work, we adapt the Gumbel tree-LSTM of Choi et al. (2018) to low-resource NMT, allowing unsupervised hierarchies to be injected into the encoder. We compare this model to sequential neural machine translation, as well as to NMT enriched with information from an external parser.

Proceedings of the Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP, pages 6–12
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

Our proposed model yields significant improvements in very low-resource NMT without requiring outside data or parsers beyond what is used in standard NMT; in addition, this model is not significantly slower to train than RNN-based models.

2 Neural Machine Translation

Neural machine translation (Cho et al., 2014; Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014) is an end-to-end neural approach to machine translation consisting of an encoder, a decoder, and an attention mechanism (Bahdanau et al., 2015). The encoder and decoder are usually LSTMs (Hochreiter and Schmidhuber, 1997) or gated recurrent units (GRUs) (Cho et al., 2014). The encoder reads in the source sentence and creates an embedding; the attention mechanism calculates a weighted combination of the words in the source sentence. This is then fed into the decoder, which uses the source representations to generate a translation in the target language.

3 Unsupervised Tree-to-Sequence NMT

We modify the standard RNN-based neural machine translation architecture by combining a sequential LSTM decoder with an unsupervised tree-LSTM encoder. This encoder induces hierarchical structure on the source sentence without syntactic supervision. We refer to models containing this encoder as *(unsupervised) tree2seq*.

In this section, we present our unsupervised tree2seq model. Section 3.1 describes the subword-level representations, while section 3.2 explains how the Gumbel tree-LSTM is used to add hierarchies in the encoder. We address top-down representations of the phrase nodes in section 3.3 and explain the attention mechanism in section 3.4.

3.1 Word Node Representations

The hierarchical encoder consists of *word nodes* (nodes corresponding to the subwords of the source sentence) and *phrase nodes* (internal nodes resulting from the induced hierarchies). In order to obtain representations of the word nodes, we run a single-layer bidirectional LSTM over the source sentence; we refer to this LSTM as the *leaf LSTM*.

3.2 Phrase Node Representation

Our proposed unsupervised tree-LSTM encoder uses a Gumbel tree-LSTM (Choi et al., 2018) to obtain the phrase nodes of the source sentence. This encoder introduces unsupervised, discrete hierarchies without modifying the maximum likelihood objective used to train NMT by leveraging straight-through Gumbel softmax (Jang et al., 2017) to sample parsing decisions.

In a Gumbel tree-LSTM, the hidden state $\mathbf{h}_p$ and memory cell $\mathbf{c}_p$ for a given node are computed recursively based on the hidden states and memory nodes of its left and right children ($\mathbf{h}_l$, $\mathbf{h}_r$, $\mathbf{c}_l$, and $\mathbf{c}_r$). This is done as in a standard binary tree-LSTM as follows:

$$\begin{bmatrix}\mathbf{i}\\ \mathbf{f}_l\\ \mathbf{f}_r\\ \mathbf{o}\\ \mathbf{g}\end{bmatrix} = \begin{bmatrix}\sigma\\ \sigma\\ \sigma\\ \sigma\\ \tanh\end{bmatrix} \left(\mathbf{W}\begin{bmatrix}\mathbf{h}_l\\ \mathbf{h}_r\end{bmatrix} + \mathbf{b}\right) \tag{1}$$

$$\mathbf{c}_p = \mathbf{f}_l \odot \mathbf{c}_l + \mathbf{f}_r \odot \mathbf{c}_r + \mathbf{i} \odot \mathbf{g} \tag{2}$$

$$\mathbf{h}_p = \mathbf{o} \odot \tanh(\mathbf{c}_p) \tag{3}$$

where $\mathbf{W}$ is the weight matrix, $\mathbf{b}$ is the bias vector, σ is the sigmoid activation function, and $\odot$ is the element-wise product.

However, the Gumbel tree-LSTM differs from standard tree-LSTMs in that the selection of nodes to merge at each timestep is done in an unsupervised manner. At each timestep, each pair of adjacent nodes is considered for merging, and the hidden states $\widehat{\mathbf{h}}_i$ for each candidate parent representation are computed using equation 3. A composition query vector $\mathbf{q}$, which is simply a vector of trainable weights, is used to obtain a score v_i for each candidate representation as follows:

$$v_i = \frac{\exp\left(\mathbf{q} \cdot \widehat{\mathbf{h}}_i\right)}{\sum_j \exp\left(\mathbf{q} \cdot \widehat{\mathbf{h}}_j\right)} \tag{4}$$

Finally, the straight-through Gumbel softmax estimator (Jang et al., 2017) is used to sample a parent from the candidates $\widehat{\mathbf{h}}_i$ based on these scores v_i; this allows us to sample a hard parent selection while still maintaining differentiability.

This process continues until there is only one remaining node that summarizes the entire sentence; we refer to this as the *root node*. At inference time, straight-through Gumbel softmax is not used; instead, we greedily select the highest-scoring candidate. See Choi et al. (2018) for a more detailed description of Gumbel tree-LSTMs.

Thus, this encoder induces a binary hierarchy over the source sentence. For a sentence of length n, there are n word nodes and $n-1$ phrase nodes (including the root node). We initialize the decoder using the root node; attention to word and/or phrase nodes is described in section 3.4.

3.3 Top-Down Encoder Pass

In the bottom-up tree-LSTM encoder described in the previous section, each node is able to incorporate local information from its respective children; however, no global information is used. Thus, we introduce a top-down pass, which allows the nodes to take global information about the tree into account. We refer to models containing this top-down pass as *top-down tree2seq* models. Note that such a top-down pass has been shown to aid in tree-based NMT with supervised syntactic information (Chen et al., 2017a; Yang et al., 2017); here, we add it to our unsupervised hierarchies.

Our top-down tree implementation is similar to the bidirectional tree-GRU of Kokkinos and Potamianos (2017). The top-down root node $\mathbf{h}^{\downarrow}_{root}$ is defined as follows:

$$\mathbf{h}^{\downarrow}_{root} = \mathbf{h}^{\uparrow}_{root} \tag{5}$$

where $\mathbf{h}^{\uparrow}_{root}$ is the hidden state of the bottom-up root node (calculated using the Gumbel tree-LSTM described in section 3.2).

For each remaining node, including word nodes, the top-down representation $\mathbf{h}^{\downarrow}_i$ is computed from its bottom-up hidden state representation $\mathbf{h}^{\uparrow}_i$ (calculated using the Gumbel tree-LSTM) and the top-down representation of its parent $\mathbf{h}^{\downarrow}_p$ (calculated during the previous top-down steps) using a GRU:

$$\begin{bmatrix} \mathbf{z}^{\downarrow}_i \\ \mathbf{r}^{\downarrow}_i \end{bmatrix} = \sigma\left(\mathbf{W}^{td}\mathbf{h}^{\uparrow}_i + \mathbf{U}^{td}\mathbf{h}^{\downarrow}_p + \mathbf{b}^{td}\right) \tag{6}$$

$$\widetilde{\mathbf{h}}^{\downarrow}_i = \tanh\left(\mathbf{W}^{td}_h\mathbf{h}^{\uparrow}_i + \mathbf{U}^{td}_h\left(\mathbf{r}^{\downarrow}_i \odot \mathbf{h}^{\downarrow}_p\right) + \mathbf{b}^{td}_h\right) \tag{7}$$

$$\mathbf{h}^{\downarrow}_i = \left(1 - \mathbf{z}^{\downarrow}_i\right)\mathbf{h}^{\downarrow}_p + \mathbf{z}^{\downarrow}_i\widetilde{\mathbf{h}}^{\downarrow}_i \tag{8}$$

where $\mathbf{W}^{td}$, $\mathbf{U}^{td}$, $\mathbf{W}^{td}_h$, and $\mathbf{U}^{td}_h$ are weight matrices; $\mathbf{b}^{td}$ and $\mathbf{b}^{td}_h$ are bias vectors; and σ is the sigmoid activation function. Note that we do not use different weights for left and right children of a given parent.

Each node needs a final representation to supply to the attention mechanism. Here, the top-down version of each node is used, because the top-down version captures both local and global information about the node.

The decoder is initialized with the top-down representation of the root node. Note, however, that this is identical to the bottom-up representation of the root node, so no additional top-down information is used to initialize the decoder. Since the root node contains information about the entire sentence, this allows the decoder to be initialized with a summary of the source sentence, mirroring standard sequential NMT.

3.4 Attention to Words and Phrases

The standard and top-down tree2seq models take different approaches to attention. The standard (bottom-up) model attends to the intermediate phrase nodes of the tree-LSTM, in addition to the word nodes output by the leaf LSTM. This follows what was done by Eriguchi et al. (2016). We use one attention mechanism for all nodes (word and phrase), making no distinction between different node types. Note that without the attention to the phrase nodes, the bottom-up tree2seq model would be almost equivalent to standard seq2seq, since the word nodes are created using a sequential LSTM (the only difference would be the use of the root node to initialize the decoder).

When the top-down pass (section 3.3) is added to the encoder, the final word nodes contain hierarchical information from the entire tree, as well as sequential information. Therefore, in the top-down tree2seq model, we attend to the top-down word nodes only, ignoring the phrase nodes. We argue that attention to the phrase nodes is unnecessary, since the word nodes summarize the phrase-level information; indeed, in preliminary experiments, attending to phrase nodes did not yield improvements.

4 Experimental Setup

4.1 Data

The models are tested on Tagalog (TL) ↔ English (EN), Turkish (TR) ↔ EN, and Romanian (RO) ↔ EN. These pairs were selected because they range from very low-resource to medium-resource, so we can evaluate the models at various settings. Table 1 displays the number of parallel training sentences for each language pair.

Language Pair	Sentences
TL↔EN	50 962
TR↔EN	207 373
RO↔EN	608 320

Table 1: Amount of parallel sentences for each language pair after preprocessing.

The TR↔EN and RO↔EN data is from the WMT17 and WMT16 shared tasks, respectively (Bojar et al., 2017, 2016). Development is done on newsdev2016 and evaluation on newstest2016. The TL↔EN data is from IARPA MATERIAL Program language collection release IARPA_MATERIAL_BASE-1B-BUILD_v1.0. No monolingual data is used for training.

The data is tokenized and truecased with the Moses scripts (Koehn et al., 2007). We use byte pair encoding (BPE) with 45k merge operations to split words into subwords (Sennrich et al., 2016). Notably, this means that the unsupervised tree encoder induces a binary parse tree over subwords (rather than at the word level).

4.2 Baselines

We compare our models to an RNN-based attentional NMT model; we refer to this model as *seq2seq*. Apart from the encoder, this baseline is identical to our proposed models. We train the seq2seq baseline on unparsed parallel data.

For translations out of English, we also consider an upper bound that uses syntactic supervision; we dub this model *parse2seq*. This is based on the mixed RNN model proposed by Li et al. (2017). We parse the source sentences using the Stanford CoreNLP parser (Manning et al., 2014) and linearize the resulting parses. We parse before applying BPE, and do not add any additional structure to segmented words; thus, final parses are not necessarily binary. This is fed directly into a seq2seq model (with increased maximum source sentence length to account for the parsing tags).

4.3 Implementation

All models are implemented in OpenNMT-py (Klein et al., 2017). They use word embedding size 500, hidden layer size 1000, batch size 64, two layers in the encoder and decoder, and dropout rate 0.3 (Gal and Ghahramani, 2016). We set maximum sentence length to 50 (150 for parse2seq source). Models are trained using Adam (Kingma and Ba, 2015) with learning rate 0.001. For tree-based models, we use a Gumbel temperature of 0.5, which performed best in preliminary experiments. The tree-LSTM component of the unsupervised tree2seq encoders has only a single layer.

BLEU	TL→EN	TR→EN	RO→EN
seq2seq	17.9	11.1	**29.3**
tree2seq	**26.1**	12.8	28.6
top-down tree2seq	25.3	**13.2**	28.6

Table 2: BLEU for the baseline and the unsupervised tree2seq systems on *→EN translation.

BLEU	EN→TL	EN→TR	EN→RO
seq2seq	15.9	8.5	27.3
parse2seq	17.1	9.0	**28.4**
tree2seq	**23.1**	9.7	27.3
top-down tree2seq	22.5	**9.8**	27.0

Table 3: BLEU for the baselines and the unsupervised tree2seq systems on EN→* translation.

We train until convergence on the validation set, and the model with the highest BLEU on the validation set is used to translate the test data. During inference, we set beam size to 12 and maximum length to 100.

5 Results

5.1 Translation Performance

Tables 2 and 3 display BLEU scores for our unsupervised tree2seq models translating into and out of English, respectively. For the lower-resource language pairs, TL↔EN and TR↔EN, the tree2seq and top-down models consistently improve over the seq2seq and parse2seq baselines. However, for the medium-resource language pair (RO↔EN), the unsupervised tree models do not improve over seq2seq, unlike the parse2seq baseline. Thus, inducing hierarchies on the source side is most helpful in very low-resource scenarios.

5.2 Unsupervised Parses

Williams et al. (2017) observed that the parses resulting from Gumbel tree-LSTMs for sentence classification did not seem to fit a known formalism. An examination of the parses induced by our NMT models suggests this as well. Furthermore, the different models (tree2seq and top-down tree2seq) do not seem to learn the same parses for the same language pair. We display example parses induced by the trained systems on a sentence from the test data in Table 4.

	Example Parse
EN→TR tree2seq	(((others have) (((dismissed him) as) a)) (j@@ (oke .)))
EN→TR top-down	((((others have) dismissed) (him as)) ((a (j@@ oke)) .))
EN→RO tree2seq	(((others have) dismissed) (him (((as a) joke) .)))
EN→RO top-down	(others (((have ((dismissed him) (as a))) joke) .))

Table 4: Induced parses on an example sentence from the test data.

Language Pair	tree2seq	top-down
EN→TL	22.1%	16.4%
EN→TR	29.3%	21.3%
EN→RO	27.2%	27.2%
TL→EN	12.7%	22.7%
TR→EN	27.7%	22.9%
RO→EN	30.8%	11.4%

Table 5: Recombined subwords in the test data.

5.3 Subword Recombination

The unsupervised parses are trained over subwords; if the induced hierarchies have a linguistic basis, we would expect the model to combine subwords into words as a first step. For each model, we calculate the percentage of subwords that are recombined correctly; the results are in Table 5. Corroborating the observations in the previous section, only a very low percentage of subwords are correctly recombined for each model. This indicates that the parses the model learns are likely not linguistic. In addition, subword recombination does not seem to correlate with translation performance.

6 Related Work

Most work on adding source hierarchical information to neural machine translation has used supervised syntax. Luong et al. (2016) used a multi-task setup with a shared encoder to parse and translate the source language. Eriguchi et al. (2016) introduced a tree-LSTM encoder for NMT that relied on an external parser to parse the training and test data. The tree-LSTM encoder was improved upon by Chen et al. (2017a) and Yang et al. (2017), who added a top-down pass. Other approaches have used convolutional networks to model source syntax. Chen et al. (2017b) enriched source word representations by extracting information from the dependency tree; a convolutional encoder was then applied to the representations. Bastings et al. (2017) fed source dependency trees into a graph convolutional encoder.

Inducing unsupervised or semi-supervised hierarchies in NMT is a relatively recent research area. Gehring et al. (2017a,b) introduced a fully convolutional model for NMT, which improved over strong sequential baselines. Hashimoto and Tsuruoka (2017) added a latent graph parser to the encoder, allowing it to learn dependency-like source parses in an unsupervised manner. However, they found that pre-training the parser with a small amount of human annotations yielded the best results. Finally, Kim et al. (2017) introduced structured attention networks, which extended basic attention by allowing models to attend to latent structures such as subtrees.

7 Conclusions

In this paper, we have introduced a method for incorporating unsupervised structure into the source side of NMT. For low-resource language pairs, this method yielded strong improvements over sequential and parsed baselines. This technique is useful for adding hierarchies to low-resource NMT when a source-language parser is not available. Further analysis indicated that the induced structures are not similar to known linguistic structures.

In the future, we plan on exploring ways of inducing unsupervised hierarchies on the decoder. Additionally, we would like to try adding some supervision to the source trees, for example in the form of pre-training, in order to see whether actual syntactic information improves our models.

Acknowledgements

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract # FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

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*. Association for Computational Linguistics.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: A case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267. 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.

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

Kehai Chen, Rui Wang, Masao Utiyama, Lemao Liu, Akihiro Tamura, Eiichiro Sumita, and Tiejun Zhao. 2017b. Neural machine translation with source dependency representation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2836–2842. 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.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29*, pages 1019–1027.

Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. 2017a. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the ACL*, pages 123–135. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017b. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1243–1252.

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*. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with Gumbel-softmax. In *5th International Conference on Learning Representations*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709. Association for Computational Linguistics.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. 2017. Structured attention networks. In *5th International Conference on Learning Representations*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the ACL*. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 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.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics.

Filippos Kokkinos and Alexandros Potamianos. 2017. Structural attention neural networks for improved sentiment analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 586–591. 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*. Association for Computational Linguistics.

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.

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.

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.

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. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

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.

Antonio Toral and Víctor M Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1063–1073. Association for Computational Linguistics.

Adina Williams, Andrew Drozdov, and Samuel R Bowman. 2017. Learning to parse from a semantic objective: It works. Is it syntax? *arXiv preprint arXiv:1709.01121*.

Baosong Yang, Derek F Wong, Tong Xiao, Lidia S Chao, and Jingbo Zhu. 2017. Towards bidirectional hierarchical representations for attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1432–1441. Association for Computational Linguistics.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2017. Learning to compose words into sentences with reinforcement learning. In *5th International Conference on Learning Representations*.