Edinburgh Research Explorer

Encoders Help You Disambiguate Word Senses in Neural Machine Translation

Citation for published version:
Tang, G, Sennrich, R & Nivre, J 2019, Encoders Help You Disambiguate Word Senses in Neural Machine Translation. in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing.* Association for Computational Linguistics (ACL), pp. 1429–1435, 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Hong Kong, Hong Kong, 3/11/19. https://doi.org/10.18653/v1/D19-1149

Digital Object Identifier (DOI):
10.18653/v1/D19-1149

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing

Encoders Help You Disambiguate Word Senses
in Neural Machine Translation

Gongbo Tang[1]   **Rico Sennrich**[2,3]   **Joakim Nivre**[1]

[1]Department of Linguistics and Philology, Uppsala University
[2]Institute of Computational Linguistics, University of Zurich
[3]School of Informatics, University of Edinburgh
`firstname.lastname@{lingfil.uu.se, ed.ac.uk}`

Abstract

Neural machine translation (NMT) has achieved new state-of-the-art performance in translating ambiguous words. However, it is still unclear which component dominates the process of disambiguation. In this paper, we explore the ability of NMT encoders and decoders to disambiguate word senses by evaluating hidden states and investigating the distributions of self-attention. We train a classifier to predict whether a translation is correct given the representation of an ambiguous noun. We find that encoder hidden states outperform word embeddings significantly which indicates that encoders adequately encode relevant information for disambiguation into hidden states. In contrast to encoders, the effect of decoder is different in models with different architectures. Moreover, the attention weights and attention entropy show that self-attention can detect ambiguous nouns and distribute more attention to the context.

1 Introduction

Neural machine translation (NMT) models (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015; Luong et al., 2015) have access to the whole source sentence for the prediction of each word, which intuitively allows them to perform word sense disambiguation (WSD) better than previous phrase-based methods, and Rios et al. (2018) have confirmed this empirically. However, it is still unclear which component dominates the ability to disambiguate word senses. We explore the ability of NMT encoders and decoders to disambiguate word senses by evaluating hidden states and investigating the self-attention distributions.

Marvin and Koehn (2018) find that the hidden states in higher encoder layers do not perform disambiguation better than those in lower layers and

conclude that encoders do not encode enough relevant context for disambiguation. However, their results are based on small data sets, and we wish to revisit this question with larger-scale data sets. Tang et al. (2018b) speculate that encoders have encoded the relevant information for WSD into hidden states before decoding but without any experimental tests.

In this paper, we first train a classifier for WSD, on a much larger data set than Marvin and Koehn (2018), extracted from ContraWSD (Rios et al., 2017), for both German→English (DE→EN) and German→French (DE→FR). The classifier is fed a representation of ambiguous nouns and a word sense (represented as the embedding of a translation candidate), and has to predict whether the two match. We can learn the role that encoders play in encoding information relevant for WSD by comparing different representations: word embeddings and encoder hidden states at different layers. We extract encoder hidden states from both RNN-based (*RNNS2S*) (Luong et al., 2015) and *Transformer* (Vaswani et al., 2017) models. Belinkov et al. (2017a,b) have shown that the higher layers are better at learning semantics. We hypothesize that the hidden states in higher layers incorporate more relevant information for WSD than those in lower layers. In addition to encoders, we also probe how much do decoder hidden states contribute to the WSD classification task.

Recently, the distributions of attention mechanisms have been used for interpreting NMT models (Ghader and Monz, 2017; Voita et al., 2018; Tang et al., 2018b; Voita et al., 2019; Tang et al., 2019). We further investigate the attention weights and attention entropy of self-attention in encoders to explore how self-attention incorporates relevant information for WSD into hidden states. As sentential information is helpful in disambiguating ambiguous words, we hypothesize that self-

attention pays more attention to the context when modeling ambiguous words, compared to modeling words in general.

Here are our findings:

- Encoders encode lots of relevant information for word sense disambiguation into hidden states, even in the first layer. The higher the encoder layer, the more relevant information is encoded into hidden states.
- Forward RNNs are better than backward RNNs in modeling ambiguous nouns.
- Decoders hidden states have different effects on WSD in *Transformer* and *RNNS2S*.
- Self-attention focuses on the ambiguous nouns themselves in the first layer and keeps extracting relevant information from the context in higher layers.
- Self-attention can recognize the ambiguous nouns and distribute more attention to the context words compared to dealing with nouns in general.

2 Methodology

2.1 WSD Classifier

ContraWSD (Rios et al., 2017) is a WSD test set for NMT. Each ambiguous noun in a specific sentence has a small number of translation candidates. We generate instances that are labelled with one candidate and a binary value indicating whether it corresponds to the correct sense.

Encoders Given an input sentence, NMT encoders generate hidden states of all input tokens. Our analysis focuses on the hidden states of ambiguous nouns (R_{ambi}). We use word embeddings from NMT models to represent the translation candidates (R_{sense}). If the ambiguous nouns or translation candidates are split into subwords, we just sum the representations.

Figure 1 illustrates the WSD classification task. We first generate hidden states for each sentence. The classifier is a feed-forward neural network with only one hidden layer. The input of the classifier is the concatenation of R_{ambi} and R_{sense}. The classifier predicts whether the translation is the correct sense of the ambiguous noun.

As the baseline, we use word embeddings from NMT models as representations of ambiguous nouns. Each ambiguous noun has only one corresponding word embedding, so such a classifier can at best learn a most-frequent-sense solution, while
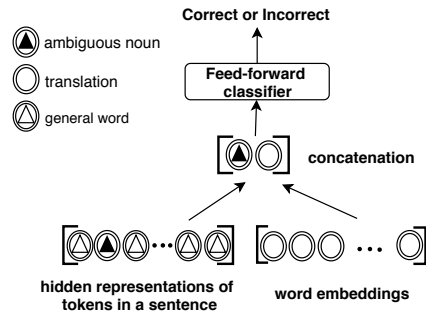

Figure 1: Illustration of the WSD classification task, using encoder hidden states to represent ambiguous nouns. The input of the classifier is the concatenation of the ambiguous word and the translation. The output of the classifier is "correct" or "incorrect".

hidden states are based on sentential information, so that ambiguous nouns have different representations in different source sentences. We can learn to what extent relevant information for WSD is encoded by encoders by comparing to the baseline.

Decoders To explore the role of decoders, we feed the decoder hidden state at the time step predicting the translation of the ambiguous noun, and the word embedding of the current translate candidate into the classifier. The decoder hidden state is extracted from the last decoder layer. To get these hidden states. we force NMT models to generate the reference translations using constrained decoding (Post and Vilar, 2018). Since decoders are crucial in NMT, we assume that the decoder hidden states incorporate more relevant information for WSD from the decoder side. Thus, we hypothesize that using decoder hidden states can achieve better WSD performance.

2.2 Attention Distribution

The attention weights can be viewed as the degree of contribution to the current word representation, which provides a way to interpret NMT models. Tang et al. (2018a) have shown that Transformers with self-attention are better at WSD than RNNs. However, the working mechanism of self-attention has not been explored. We try to use the attention distributions in different encoder layers to interpret how self-attention incorporates relevant information to disambiguate word senses.

All the ambiguous words in the test set are nouns. Ghader and Monz (2017) have shown that nouns have different attention distributions from other word types. Thus, we compare the attention distributions of ambiguous nouns to nouns in gen-

eral[1] in two respects. One is the attention weight over the word itself. The other one is the concentration of attention distributions. We use attention entropy (Ghader and Monz, 2017) to measure the concentration.

$$E_{At}(x_t) = -\sum_{i=1}^{|x|} At(x_i, x_t) \log At(x_i, x_t) \quad (1)$$

Here x_i denotes the ith source token, x_t is the current source token, and $At(x_i, x_t)$ represents the attention weight from x_t to x_i. We merge subwords after encoding, following the method in Koehn and Knowles (2017).[2] Each self-attention layer has multiple heads and we average the attention weights from all the heads.

In theory, sentential information is more important for ambiguous words that need to be disambiguated than non-ambiguous words. From the perspective of attention weights, for ambiguous words, we hypothesize that self-attention distributes more attention to the context words to capture the relevant sentential information, compared to words in general. From the perspective of attention entropy, we hypothesize that self-attention focuses on the related context words rather than the entire sentence which produces a smaller entropy. If ambiguous words have a lower weight and a smaller entropy than words in general, the results can confirm our hypotheses.

3 Experiments

For NMT models, we use the *Sockeye* (Hieber et al., 2017) toolkit to train *RNNS2S*s and *Transformer*s. DE→EN training data is from the WMT17 shared task (Bojar et al., 2017). DE→FR training data is from Europarl (v7) (Koehn, 2005) and News Commentary (v11) cleaned by Rios et al. (2017).[3] In *ContraWSD*, each ambiguous noun has a small number of translation candidates. The average number of word senses per noun is 2.4 and 2.3 in DE→EN and DE→FR, respectively. We generate instances that are labelled with one candidate and a binary value indicating whether it corresponds to the correct sense. we get 50,792 and 43,268 instances in DE→EN and DE→FR, respectively. 5K/5K examples are randomly selected as the test/development set. The remaining

---

[1]We use *TreeTagger* (Schmid, 1995) to identify nouns.
[2](1) If a query word is split into subwords, we add their attention weights. (2) If a key word is split into subwords, we average their attention weights.
[3]http://data.statmt.org/ContraWSD/

examples are used for training. We train 10 times with different seeds for each classifier and apply average accuracy. Table 1 lists the detailed statistics of the data. More experimental details are provided in the Appendix.

	DE→EN	DE→FR
NMT training data	5.9M	2.1M
Word senses	84	71
Lexical ambiguities	7,359	6,746
Instances	50,792	43,268

Table 1: Training data for NMT, and data extracted from ContraWSD: Word senses: total number of senses. Lexical ambiguities: number of sentences containing an ambiguous word. Instances: number of instances generated for WSD classification.

3.1 Results

Table 2 provides the BLEU scores and the WSD accuracy on test sets, using different representations to represent ambiguous nouns. *ENC* denotes encoder hidden states; *DEC* means decoder hidden states.

	DE→EN		DE→FR	
	RNN.	*Trans.*	*RNN.*	*Trans.*
BLEU	29.1	32.6	17.0	19.3
Embedding	63.1	63.2	68.7	68.9
ENC	94.2	97.2	91.7	95.6
DEC	97.9	91.2	95.1	91.6

Table 2: BLEU scores of NMT models, and WSD accuracy on the test set using word embeddings or hidden states to represent ambiguous nouns. The hidden states are from the highest layer.[4] *RNN.* and *Trans.* denote *RNNS2S* and *Transformer* models, respectively.

ENC achieves much higher accuracy than *Embedding*. The WSD accuracy of *Embedding* are around 63% and 69% in the two languages. While the accuracy of *ENC* increases to over 91%. The absolute accuracy gap varies from 23% to 34%, which is substantial. This result indicates that encoders have encoded a lot of relevant information for WSD into hidden states. In addition, *DEC* achieves even higher accuracy than *ENC* in *RNNS2S* models but not in *Transformer* models.

---

[4]For encoders in *RNNS2S*s, this is the last backward RNN.

4 Analysis

4.1 WSD Classification

4.1.1 RNNS2S vs. Transformer

*RNNS2S*s are inferior to *Transformer*s distinctly in BLEU score. However, the hidden states from *RNNS2S* also improve accuracy significantly, just not as much as those from *Transformer* models. This result indicates that *Transformer*s encode more relevant context for WSD than *RNNS2S*s and accords with the finding in Tang et al. (2018a) that *Transformer*s perform WSD better than *RNNS2S*s.

The results of *ENC* using *RNNS2S* in Table 2 are only based on hidden states from the last backward RNN. We also concatenate the hidden states from both forward and backward RNNs and get higher accuracy, 96.8% in DE→EN and 95.7% in DE→FR. The WSD accuracy of using bidirectional hidden states are competitive to using hidden states from Transformer models. However, concatenating forward and backward hidden states doubles the dimension. Thus, the comparison is not completely fair.

4.1.2 Encoder Depth

Figure 2 illustrates WSD accuracy in different encoder layers, with standard deviation as error bars. Even the hidden states from the first layer boost the WSD performance substantially compared to using word embeddings. This means that most of the relevant information for WSD has been encoded into hidden states in the first encoder layer. For *Transformer*s, the WSD accuracy goes up consistently as the encoder layer gets higher. *RNNS2S* has 3 stacked bi-directional RNNs. Both forward and backward layers get higher accuracy when the depth increases. All the models show that hidden states in higher layers incorporate more relevant information for WSD.

Our results conflict with the findings in Marvin and Koehn (2018) where they find that hidden states in higher encoder layers do not perform disambiguation better than those in lower layers. One of the distinct differences from Marvin and Koehn (2018) is that we train the classifier with ∼40K instances. While they employ 426 examples. Moreover, they extract encoder hidden states from NMT models with different layers rather than different layers of the same model.

Moreover, it is interesting that the forward layers surpass the backward layers in the same bidirectional RNN. One possible explanation is that
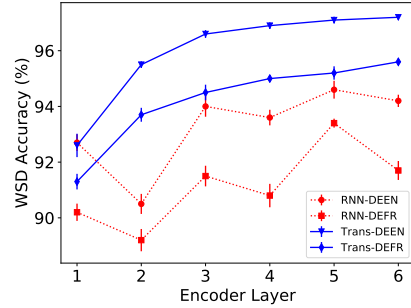

Figure 2: The WSD accuracy of using hidden states in different encoder layers, with standard deviation as error bars. For *RNNS2S*s, the odd layers (1, 3, 5) are forward RNNs and the even layers (2, 4, 6) are backward RNNs.

there is more relevant information for WSD before ambiguous nouns rather than after ambiguous nouns, which makes forward RNNs inject more relevant information into the hidden states of ambiguous nouns than backward RNNs.

4.1.3 Decoders

As Table 2 shows, RNN decoder hidden states could further improve the classification accuracy which accords with our hypothesis. It implies that the relevant information for WSD in the target-side has been well incorporated into the decoder hidden states to predict the translations of ambiguous nouns. It is curious that Transformer decoder hidden states are inferior to Transformer encoder hidden states in our WSD classification task, given that Tang et al. (2018a) and Rios et al. (2018) report better results with contrastive evaluation and semi-automatic evaluation of 1-best translations for Transformer models than for *RNNS2S*. However, note that our evaluation merely tests whether the information necessary for word sense disambiguation is encoded in hidden states and can be extracted by our binary classifier. In practice, decoder hidden states are used for predicting a target word from the entire vocabulary, and thus need to encode additional information which may confound our classifier.

Despite these differences between *RNNS2S* and the Transformer, our results show that WSD is already possible on the basis of the encoder representation of the ambiguous noun, and that extracting contextual information via encoder-decoder attention or from the target history is not essential for WSD.

4.2 Self-attention

4.2.1 Attention Weights

Figure 3 exhibits the average attention weights of ambiguous nouns and all nouns over themselves in different layers. In the first layer, the attention weights are distinctly higher than those in higher layers. 87% and 90% of ambiguous nouns assign the highest attention to themselves in DE→EN and DE→FR, respectively. The attention weights drop dramatically from the second layer. It thus seems that self-attention pays more attention to the ambiguous nouns themselves in the first layer and to context words in the following layers.
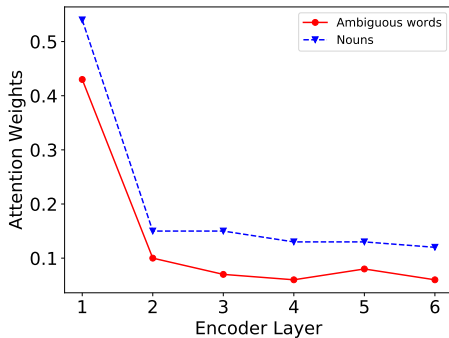

Figure 3: The average attention weights of ambiguous nouns and general nouns over themselves in different layers, in DE→EN (same pattern in DE→FR).

The attention weights of ambiguous nouns are lower than those of nouns in general. That is, more attention is distributed to context words, which implies that self-attention recognizes ambiguous nouns and distributes more attention to the context. We can conclude that self-attention pays more attention to context words to extract relevant information for disambiguation in all the layers, compared to nouns in general.

4.2.2 Attention Entropy

Section 4.2.1 has shown that self-attention of ambiguous nouns distributes more attention to the context than self-attention of nouns in general but what does the attention distribution look like? Figure 4 displays the average attention entropy of ambiguous nouns and all nouns in different layers. From the second layer, ambiguous nouns have smaller attention entropy than nouns in general, which means that self-attention mainly distributes attention to some specific words rather than all the words. As self-attention focuses on the ambiguous nouns themselves in the first layer, this result accords with our hypothesis as well.
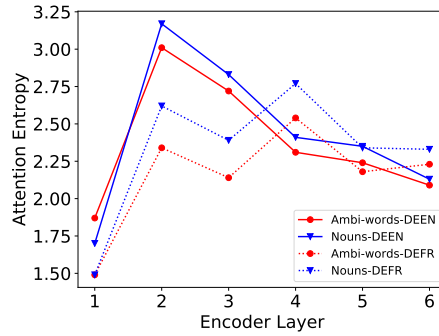

Figure 4: The average attention entropy of ambiguous nouns and nouns in different encoder layers.

In addition, there is a roughly general pattern that the attention entropy first rises and then drops. A plausible explanation is that the attention entropy first rises because context information is extracted from the entire sentence and later drops due to focusing on the most relevant context tokens.

5 Conclusion

In this paper, we investigate the ability of NMT encoders and decoders to disambiguate word senses. We first train a neural classifier to predict whether the translation is correct given the representations of ambiguous nouns. We find that encoder hidden states outperform word embeddings significantly in the classification task which indicates that relevant information for WSD has been well integrated by encoders. In addition, the higher the encoder layer, the more relevant information is encoded into hidden states. Moreover, the effect of decoder hidden states on WSD is different in Transformer and *RNNS2S* models.

We further explore the attention distributions of self-attention in encoders. The results show that self-attention can detect ambiguous nouns and distribute more attention to context words. Besides, self-attention focuses on the ambiguous nouns themselves in the first layer, then keeps extracting features from context words in higher layers.

References

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

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. 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, Vancouver, Canada. Association for Computational Linguistics.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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.

Kyunghyun Cho, Bart van Merrienboer, 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, Doha, Qatar. Association for Computational Linguistics.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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

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, Seattle, Washington, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California, USA.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit*, pages 79–86, Phuket, Thailand.

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, Vancouver, Canada. Association for Computational Linguistics.

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.

Rebecca Marvin and Phillip Koehn. 2018. Exploring word sense disambiguation abilities of neural machine translation systems. In *Proceedings of AMTA 2018 (Volume 1: MT Research Track)*, pages 125–131, Boston, USA. Association for Machine Translation in the Americas.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Annette Rios, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19, Copenhagen, Denmark. Association for Computational Linguistics.

Annette Rios, Mathias Müller, and Rico Sennrich. 2018. The word sense disambiguation test suite at WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 588–596, Belgium, Brussels. Association for Computational Linguistics.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*, Dublin, Ireland. 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.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Neural Information Processing Systems 2014*, pages 3104–3112, Montréal, Canada.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018a. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272, Brussels, Belgium. Association for Computational Linguistics.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018b. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 26–35, Belgium, Brussels. Association for Computational Linguistics.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2019. Understanding neural machine translation by simplification: The case of encoder-free models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2019*, Varna, Bulgaria.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Data

NMT For DE→EN, we use *newstest2013* as the validation set, and use *newstest2017* as the test set. For DE→FR, we use *newstest2013* as the evaluation set, and use *newstest2012* as the test set.

A.2 Experimental Settings

NMT We implemented *RNNS2S* models with stacked bi-directional RNNs and implemented the self-attention in Transformer encoders to output the attention distributions. We use *Adam* (Kingma and Ba, 2015) as the optimizer. The initial learning rate is set to 0.0002. All the neural networks have 6 layers.[5] The size of embeddings and hidden units is 512. The attention mechanism in *Transformer* has 8 heads. We learn a joint BPE model with 32,000 subword units (Sennrich et al., 2016). All BLEU scores are computed with *SacreBLEU* (Post, 2018).

WSD Classification The classifiers are feed-forward neural networks with only one hidden layer, using ReLU non-linear activation. The size of the hidden layer is set to 512. We use Adam learning algorithm as well with mini-batches of size 3,000. The classifiers are trained using a cross-entropy loss. Each classifier is trained for 80 epochs[6] and the one performs best on the development set is selected for evaluation.

---

[5]The RNN encoder is a stack of three bi-directional RNNs which is equivalent to 6 uni-directional RNNs.

[6]The classifiers fed decoder states are trained 200 epochs to converge.