An Analysis of Attention Mechanisms: The Case of Word Sense Disambiguation in Neural Machine Translation

Gongbo Tang1  Rico Sennrich2,3  Joakim Nivre1

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

Abstract
Recent work has shown that the encoder-decoder attention mechanisms in neural machine translation (NMT) are different from the word alignment in statistical machine translation. In this paper, we focus on analyzing encoder-decoder attention mechanisms, in the case of word sense disambiguation (WSD) in NMT models. We hypothesize that attention mechanisms pay more attention to context tokens when translating ambiguous words. We explore the attention distribution patterns when translating ambiguous nouns. Counter-intuitively, we find that attention mechanisms are likely to distribute more attention to the ambiguous noun itself rather than context tokens, in comparison to other nouns. We conclude that attention is not the main mechanism used by NMT models to incorporate contextual information for WSD. The experimental results suggest that NMT models learn to encode contextual information necessary for WSD in the encoder hidden states. For the attention mechanism in Transformer models, we reveal that the first few layers gradually learn to “align” source and target tokens and the last few layers learn to extract features from the related but unaligned context tokens.

1 Introduction
Human languages exhibit many different types of ambiguity. Lexical ambiguity refers to the fact that words can have more than one semantic meaning. Dealing with these lexical ambiguities is a challenge for various NLP tasks. Word sense disambiguation (WSD) is recognizing the correct meaning of an ambiguous word, with the help of contextual information.

In statistical machine translation (SMT) (Koehn et al., 2003), a system could explicitly take context tokens into account to improve the translation of ambiguous words (Vickrey et al., 2005). By contrast, in neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014), especially in attentional NMT (Bahdanau et al., 2015; Luong et al., 2015), each hidden state incorporates contextual information. Hence, NMT models could potentially perform WSD well. However, there are no empirical results to indicate that the hidden states encode the contextual information needed for disambiguation. Moreover, how the attention mechanism deals with ambiguous words is also not known yet.

In this paper, we focus on the question of how encoder-decoder attention mechanisms deal with ambiguous nouns. We explore two different attention mechanisms. One is the vanilla one-layer attention mechanism (Bahdanau et al., 2015; Luong et al., 2015), and the other one is the Transformer attention mechanism (Vaswani et al., 2017).

Rios et al. (2017) find that attentional NMT models perform well in translating ambiguous words with frequent senses, while Liu et al. (2018) show that there are plenty of incorrect translations of ambiguous words. In Section 4, we evaluate the translations of ambiguous nouns, using the test set from Rios et al. (2017). In this setting, we expect to get a more accurate picture of the WSD performance of NMT models.

In Section 5, we present a fine-grained investigation of attention distributions of different attention mechanisms. We focus on the process of translating the given ambiguous nouns. Previous studies (Ghader and Monz, 2017; Koehn and Knowles, 2017) have shown that attention mechanisms learn to pay attention to some unaligned but useful context tokens for predictions. Thus, we hypothesize that attention mechanisms distribute more attention to context tokens when translating
ambiguous nouns, compared to when translating other words. To test this hypothesis, we compare the attention weight over ambiguous nouns with the attention weight over all words and all nouns.

In Section 6, we first compare the two different attention mechanisms. Then, we explore the relation between accuracy and attention distributions when translating ambiguous nouns. In the end, we investigate the error distributions over frequency.

Our main findings are summarized as follows:

- We find that WSD is challenging in NMT, and data sparsity is one of the main issues.
- We show that attention mechanisms prefer to pay more attention to the ambiguous nouns rather than context tokens when translating ambiguous nouns.
- We conclude that encoder-decoder attention is not the main mechanism used by NMT models to incorporate contextual information for WSD. Experimental results suggest that models learn to encode contextual information necessary for WSD in the encoder hidden states.
- We reveal that the attention mechanism in Transformers first gradually learns to extract features from the “aligned” source tokens. Then, it learns to capture features from the related but unaligned source context tokens.

2 Related Work

Both Rios et al. (2017) and Liu et al. (2018) propose some techniques to improve the translation of ambiguous words. Rios et al. (2017) use sense embeddings and lexical chains as additional input features. Liu et al. (2018) introduce an additional context vector. There is an apparent difference in evaluation between these two studies. Rios et al. (2017) design a constrained WSD task. They create well-designed test sets to evaluate the performance of NMT models in distinguishing different senses of ambiguous words, rather than evaluating the translations of ambiguous words directly. By contrast, Liu et al. (2018) evaluate the translations of ambiguous words but on a common test set. Scoring the contrastive translations is not evaluating the real output of NMT models. In this paper, we directly evaluate the translations generated by NMT models, using ContraWSD as the test set.

In NMT, the encoder may encode contextual information into the hidden states. Marvin and Koehn (2018) explore the ability of hidden states at different encoder layers in WSD, while we focus on exploring the attention mechanisms that connect the encoder and the decoder.

Koehn and Knowles (2017) and Ghader and Monz (2017) investigate the relation between attention mechanisms and the traditional word alignment. They find that attention mechanisms not only pay attention to the aligned source tokens but also distribute attention to some unaligned source tokens. In this paper, we perform a more fine-grained investigation of attention mechanisms, focusing on the task of translating ambiguous nouns. We also explore the advanced attention mechanisms in Transformer models (Vaswani et al., 2017).

The encoder-decoder attention mechanisms differ in NMT models. Tang et al. (2018b) evaluate different NMT models, but focusing on NMT architectures. Tang et al. (2018a); Domhan (2018) compare different attention mechanisms. However, there is no detailed analysis on attention mechanisms.

In this paper, we mainly investigate the encoder-decoder attention mechanisms. More specifically, we explore how attention mechanisms work when translating ambiguous nouns.

3 Background

3.1 Attention Mechanisms

Attention mechanisms were initially proposed to learn the alignment between source and target tokens by Bahdanau et al. (2015) and Luong et al. (2015), in order to improve the performance of NMT. However, attention mechanisms are different from the traditional word alignment in SMT which learns the hard alignment between source and target tokens. Attention mechanisms learn to extract features from all the source tokens when generating a target token. They assign weights to all the hidden states of source tokens. The more related hidden states are assigned larger weights. Then attention mechanisms feed a context vector \( c_t \), which is extracted from the encoder, into the decoder for target-side predictions.

We use \( h \) to represent the hidden state set \( \{ h_1, h_2, \ldots, h_n \} \) in the encoder, where \( n \) is the number of source-side tokens. Then \( c_t \) is computed by Equation 1:

\[
c_t = \alpha_t h
\]  

where \( \alpha_t \) is the attention vector at time step \( t \). \( \alpha_t \) is
a normalized distribution of a score computed by
the hidden state set \( h \) and the decoder state \( s_{t-1} \),
as described by Equation 2:
\[
a_t = \text{softmax}(\text{score}(s_{t-1}, h))
\]  
(2)

There are different \( \text{score}() \) functions to compute
the attention vector \( a_t \), including multi-layer percep-
tron (MLP), dot product, multi-head attention,
etc. In this paper, the vanilla attention mechanism
employs MLP. The advanced attention mechanism
applies multi-head attention with scaled dot prod-
uct, which is the same as the attention mechanism
in Transformer (Vaswani et al., 2017).

Figure 1 illustrates different attention mecha-
nisms. In vanilla attention mechanisms (Bah-
danau et al., 2015; Luong et al., 2015), the
context vector \( c_t \) is only fed into the first layer of
the decoder networks. Then the single- or multi-
layer decoder networks compute from bottom to
top to predict target tokens. The vanilla attention
mechanisms can only extract the source-side fea-
tures once, which may be insufficient. Therefore,
Gehring et al. (2017) and Vaswani et al. (2017)
feed a context vector into each decoder layer. The
higher layer could take the result of the previous
layer into account when computing the new atten-
tion. More recently, Domhan (2018) has shown
that multi-layer attention is crucial in NMT mod-
els. Moreover, Vaswani et al. (2017) also propose
the multi-head attention mechanism. In contrast to
the single-head attention, there are multiple atten-
tion functions which compute the attention from
the linearly projected vectors in parallel. Then,
the context vectors from all the heads are concate-
nated and fed into the decoder networks.

3.2 ContraWSD

ContraWSD\(^3\) from Rios et al. (2017) consists of
contrastive translation sets where the human ref-

\(^3\)https://github.com/a-rios/ContraWSD

currence translations are paired with one or more
contrastive variants. Given an ambiguous word in
the source sentence, the correct translation is re-
placed by an incorrect translation corresponding
to another meaning of the ambiguous word. For
example, in a case where the English word ‘line’
is the correct translation of the German source
word ‘Schlange’, ContraWSD replaces ‘line’ with
other translations of ‘Schlange’, such as ‘snake’ or
‘serpent’, to generate contrastive translations. To
evaluate the performance on disambiguation, con-
trastive translations are designed not to be easily
identified as incorrect based on grammatical and
phonological features.

ContraWSD is extracted from a large amount of
balanced parallel text. It contains 84 different Ger-
man word senses. It has 7,200 German→English
lexical ambiguities and each lexical ambiguity in-
stance has 3.5 contrastive translations on average.
All the ambiguous words are nouns so that the
WSD is not simply based on syntactic context.

4 Evaluation

Instead of using NMT models to score the con-
trastive translations, we use NMT models to trans-
late source sentences and evaluate the translations
of the ambiguous nouns directly. We evaluate
two popular NMT models with different attention
mechanisms. One is RNNS2S with the vanilla at-
tention mechanism, and the other is Transformer
with the advanced attention mechanism.

We apply fast-align (Dyer et al., 2013) to get
the aligned translations of ambiguous nouns. To
achieve better alignment, we run fast-align on both
training data and test data which includes refer-
cence translations and generated translations. How-
ever, for some ambiguous nouns, there is no align-
ment. We call these ambiguous nouns filtered.

There are multiple reference translations for

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ n \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ h \]

\[ a_t \]

\[ s_{t-1} \]

\[ c_t \]

\[ n \]

\[ n \]-layer

blocks

\[ a_t \]
each ambiguous noun in ContraWSD. We additionally add their synonyms\(^4\) into the reference translations as well. The non-reference translations are crawled from the Internet\(^5\).

In addition to the filtered nouns, the translations of the ambiguous nouns are classified into six groups, depending on which class (references, incorrect senses, no translation) the translations at aligned/unaligned positions belong to, as described in Table 1. For instance, in C3, there is neither a correct nor an incorrect sense at the aligned position. However, there is a reference translation at an unaligned position.

Table 1: Different groups of translations. Ref. denotes the reference translations. Incor. represents the incorrect senses. No means that there is neither a correct nor an incorrect sense of the ambiguous noun. √ indicates that the translations belong to the reference translations or incorrect senses or neither.

<table>
<thead>
<tr>
<th>Group</th>
<th>Aligned</th>
<th>Unaligned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ref.</td>
<td>Incor.</td>
</tr>
<tr>
<td>C1</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>W1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drop</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Since the alignment learnt by fast-align is not perfect, we also consider the translations at unaligned positions. All the translations in C1, C2, C3 groups are viewed as correct translations. Thus, the accuracy of an NMT model on this test set is the amount of translations in Group C1, C2, C3, divided by the sum of ambiguous noun instances. Formally, \(\text{Accuracy} = (C1 + C2 + C3)/(C1 + C2 + C3 + W1 + W2 + Drop + Filtered)\), where C1, C2, W1, C3, W2, Drop, and Filtered are the amount of translations in each group.

### 4.1 Experimental Settings

We use the Sockeye (Hieber et al., 2017) toolkit, which is based on MXNet (Chen et al., 2015), to train models. In addition, we have extended Sockeye to output the distributions of encoder-decoder attention in Transformer models, from different attention heads and different attention layers.

All the models are trained with 2 GPUs. During training, each mini-batch contains 4096 tokens. A model checkpoint is saved every 4,000 updates. We use Adam (Kingma and Ba, 2015) as the optimizer. The initial learning rate is set to 0.0002. If the performance on the validation set has not improved for 8 checkpoints, the learning rate is multiplied by 0.7. We set the early stopping patience to 32 checkpoints. All the neural networks have 8 layers. For RNNs2S, the encoder has 1 bi-directional LSTM and 6 stacked uni-directional LSTMs, and the decoder is a stack of 8 uni-directional LSTMs. The size of embeddings and hidden states is 512. We apply layer-normalization and label smoothing (0.1) in all models. We tie the source and target embeddings. The dropout rate of embeddings and Transformer blocks is set to 0.1. The dropout rate of RNNs is 0.2. The attention mechanism in Transformer has 8 heads.

We use the training data from the WMT17 shared task.\(^6\) We choose newstest2013 as the validation set, and use newstest2014 and newstest2017 as the test sets. All the BLEU scores are measured by SacreBLEU. There are about 5.9 million sentence pairs in the training set after preprocessing with Moses scripts. We learn a joint BPE model with 32,000 subword units (Sennrich et al., 2016). There are 6,330 sentences left after filtering the sentences with segmented ambiguous nouns. We employ the models that have the best perplexity on the validation set for the evaluation.

### 4.2 Results

Table 2 gives the performance of NMT models on newstests and ContraWSD. The detailed translation distributions over different groups are also provided. Transformer is much better than RNNs2S in both newstests and ContraWSD. Compared to the accuracy of scoring contrastive translation pairs (Score), the accuracy of evaluating the translations (Acc.) is apparently lower.

There are 8–10% of ambiguous nouns belonging to Drop and Filtered for both models. We manually checked the translations of sentences with these ambiguous nouns and found that 250 and 206 ambiguous nouns (41%) are translated correctly by RNNs2S and Transformer, respectively. Our automatic classification failed for two reasons. On the one hand, because the models are trained at subword-level, there are a lot of subwords in the translations. The correctly gener-

\(^4\)Synonyms from WordNet (Miller, 1995)
\(^5\)https://www.linguee.com/german-english
\(^6\)http://www.statmt.org/wmt17/translation-task.html
<table>
<thead>
<tr>
<th>Model</th>
<th>2014</th>
<th>2017</th>
<th>C1</th>
<th>C2</th>
<th>W1</th>
<th>C3</th>
<th>W2</th>
<th>Drop</th>
<th>Filtered</th>
<th>Acc.</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNS2S</td>
<td>23.3</td>
<td>25.1</td>
<td>4,560</td>
<td>187</td>
<td>863</td>
<td>81</td>
<td>31</td>
<td>333</td>
<td>275</td>
<td>76.27</td>
<td>84.01</td>
</tr>
<tr>
<td>Transformer</td>
<td>26.7</td>
<td>27.5</td>
<td>4,982</td>
<td>140</td>
<td>599</td>
<td>85</td>
<td>23</td>
<td>308</td>
<td>193</td>
<td>82.26</td>
<td>90.34</td>
</tr>
</tbody>
</table>

Table 2: Evaluation results of NMT models and the distributions of translations. 2014 and 2017 denote the BLEU scores on newstest2014 and newstest2017. Acc. (in %) is short for accuracy. Score (in %) is the accuracy using NMT models to score contrastive translation pairs. Filtered is the amount of translations that there is no learnt alignment for the ambiguous nouns.

5 Ambiguous Nouns in Attentional NMT

Ghader and Monz (2017) show that there are different attention patterns for words of different part-of-speech (POS) tags, which sheds light on interpreting attention mechanisms. In this section, we investigate the attention distributions over source-side ambiguous nouns.

5.1 Hypothesis and Tests

Attention mechanisms not only pay attention to the hidden states at aligned positions but also distribute attention to the hidden states at unaligned positions. The hidden states at unaligned positions can influence the generation of the current token. In general, NLP models disambiguate ambiguous words by means of context words. Thus, for ambiguous nouns, we hypothesize that attention mechanisms distribute more attention to context tokens for disambiguation.

We test our hypothesis via two different comparisons. We use \( w_{ambi} \) to denote the average attention weight over the ambiguous nouns and employ \( w_{nouns} \) to represent the average attention weight over all nouns \(^7\) (including the ambiguous nouns), while \( w_{tokens} \) denotes the average attention weight over all tokens. \(^8\) We first compare \( w_{ambi} \) with \( w_{tokens} \). As nouns have a more集中d attention distribution than other word types (Ghader and Monz, 2017), we then compare \( w_{ambi} \) with \( w_{nouns} \). If \( w_{ambi} \) is the smallest, it supports our hypothesis.

The NMT models we evaluated are trained at subword-level. When we compute the attention distributions, we only consider the ambiguous nouns that are not segmented into subwords. To some extent, we therefore conduct an analysis of frequent tokens. We employ the alignment learnt by fast-align to find the step of translating the current source token.

Given the attention distribution matrix \( M \in \mathbb{R}^{l_s \times l_t} \) of a sentence translation, \( l_t \) represents the length of the target sentence, while \( l_s \) denotes the length of the source sentence. Each column is the attention distribution over all the source tokens when generating the current target token. Each row is the attention distribution over the current source token at all the translation steps. \( w \) represents the attention weight over any tokens. If the \( i \)th source token is aligned to the \( j \)th target token, then \( w = [M]_{ij} \). If a token is aligned to more than one token, we choose the largest attention weight as \( w \). \(^9\)

As for Transformer attention mechanisms, there are multiple layers, and each layer has multiple heads. We maximize the attention weights in different heads to represent the attention distribution matrix for each attention layer. \(^10\) We first compute \( w_{ambi}, w_{nouns}, \) and \( w_{tokens} \) for each attention layer. Then we average these weights.

5.2 Results

As Table 3 shows, \( w_{ambi} \) is substantially larger than \( w_{tokens} \) in both models. Even though \( w_{nouns} \) is much larger compared to \( w_{tokens}, w_{ambi} \)

\(^7\)We use the TreeTagger (Schmid, 1999) to tag German.
\(^8\)Subword tokens are excluded, which account for 32%.
\(^9\)A source token may be aligned to a set/subset of subword sequences, but the attention mechanism only assigns the corresponding weight to one of the subwords. We select the maximal weight rather than the average weight.
\(^10\)We visualize both the maximal and average attention weights. We find that maximal attention weights are more representative in feature extraction.
is still greater than $w_{nouns}$, especially in Transformer. This result is against our hypothesis. That is to say, attention mechanisms do not distribute more attention to context tokens when translating an ambiguous noun. Instead, attention mechanisms pay more attention to the ambiguous noun itself. We assume that the contextual information has already been encoded into the hidden states by the encoder, and attention mechanisms do not learn which source words are useful for WSD.

<table>
<thead>
<tr>
<th>Model</th>
<th>$w_{ambi}$</th>
<th>$w_{tokens}$</th>
<th>$w_{nouns}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN S2S</td>
<td>0.63</td>
<td>0.48</td>
<td>0.62</td>
</tr>
<tr>
<td>Transformer</td>
<td>0.74</td>
<td>0.57</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 3: Average attention weights over ambiguous nouns, non-subword tokens, and nouns.

Figure 2 demonstrates the average attention weights of the ambiguous nouns, nouns, and non-subword tokens in different Transformer attention layers. In each attention layer, $w_{ambi}$ is always the largest attention weight. It is very interesting that the attention weights keep increasing at lower layers and achieve the largest weight at Layer 5. Then $w_{tokens}$ decreases steadily, while $w_{ambi}$ and $w_{nouns}$ have a distinct drop in the final attention layer. We also re-train a model with 6 attention layers, and we get a figure with the same pattern, but the largest attention weights appear at Layer 4. We will give a further analysis of Transformer attention mechanisms in Section 6.1.

![Average attention weights of ambiguous nouns, nouns, and non-subword tokens in different Transformer attention layers.](image)

6 Analysis

We first give our analysis of the two different attention mechanisms based on the attention distributions and visualizations. Then, we explore the relation between translation accuracy and attention weight over the ambiguous nouns. In the end, we provide the error distributions over frequency.

6.1 Vanilla Attention vs. Advanced Attention

As Table 2 shows, the Transformer model with advanced attention mechanisms is distinctly better than the RNN model with vanilla attention mechanisms. Even though there are differences in the encoder and decoder networks, we focus on the comparison between these two attention mechanisms. Moreover, there is no existing empirical interpretation of the advanced attention mechanisms.

Figure 3 demonstrates the attention distributions of different models when translating ambiguous nouns. For the vanilla attention mechanism in the RNN model, most of the attention weights are relatively uniformly distributed in $[0.5, 0.9)$. While the patterns in advanced attention mechanisms are completely different. In the first layer, most of the attention weights are smaller than 0.1. The larger attention weights, the fewer instances, except when the weight is larger than 0.9. In the following layers, the attention weights are getting more and more concentrated in $[0.9, 1)$ until the fifth layer. After the fifth layer, the amount in $[0.9, 1)$ decreases dramatically. We hypothesize that the first few layers are learning the “alignment” gradually. When attention mechanisms finish the “alignment” learning, they start to capture contextual features from the related but unaligned context tokens. In the last layer, the attention is almost equally distributed over all the attention ranges except $(0, 0.1)$. That is to say, for some ambiguous nouns, the weights are large. For the other ambiguous nouns, the weights are small. It indicates that there is no clear attention distribution pattern over ambiguous nouns in the last layer.

Figure 4 shows the average attention weights over word tokens and subword tokens ($w_{subwords}$). In the first five layers, $w_{subwords}$ is clearly lower than $w_{tokens}$ which can be taken to show that attention mechanisms focus on the “alignment” of single word tokens, while $w_{subwords}$ surpasses $w_{tokens}$ from the sixth layer. We conclude that attention mechanisms focus on subwords instead of word tokens. Many words are segmented into multiple consecutive subwords and not all the subwords are aligned to the expected target tokens. Thus, the pattern over subword tokens demonstrates that attention mechanisms are learning to capture context-level features.
We further validate the hypothesis by visualizing the attention distributions. Table 4 demonstrates the visualization of attention distributions of different attention mechanisms.

‘Stelle’ is an ambiguous noun, whose reference translations are ‘job/position/work’. ‘Stelle’ also has other translations such as ‘location/spot/site’. The context tokens ‘garantiert’ (guarantee) and ‘Leuten’ (people) contribute to disambiguating ‘Stelle’. However, the RNN model could translate ‘Stelle’ correctly but only pays a little attention to ‘Leuten’.

In the first layer, the attention mechanism does not pay attention to the correct source tokens if we only consider the larger attention weights. Then the “alignment” is learnt gradually in the following layers. The attention mechanism could pay attention to all the correct source tokens in the fifth layer. In addition, the attention mechanism could learn to pay attention to the related but unaligned source tokens in the eighth layer. For instance, the attention mechanism also attends to ‘Stelle’ when generating ‘guarantees’, and attends to ‘garantiert’ and ‘Leuten’ when generating ‘job’. These source tokens are not clearly attended to in the fifth layer.

Since the vanilla attention mechanism is only one layer with one head, it does not perform as well as the advanced attention mechanism in learning to pay attention to context tokens. For instance, the attention mechanism in RNN only distributes a little attention to ‘Leuten’ when generating ‘job’.

6.2 Accuracy and Attention Weights

We explore the relation between WSD accuracy and the attention weights over ambiguous nouns. As the alignment learnt by fast-align does not guarantee that each ambiguous noun is aligned to the corresponding translation, we only consider the translations belonging to Group CI, WI, and Drop. Figure 5 shows the WSD accuracy over different attention ranges. Obviously, the accuracy is higher when the attention weight is greater.
Table 4: An example of attention visualization (German→English). Each row is the attention distribution over all the source tokens at each time step. Each column represents the attention weight over a source token at all the time steps. Layer 1 to Layer 8 are attention layers in the Transformer model. Each attention layer has 8 heads, and the attention weights in each row are the maximal of all the heads. Thus, the summation of attention weights in each row is larger than 1. Darker blue means larger attention weights.

result further confirms our assumption in Section 5 that the contextual information for disambiguation has been learnt by the encoder. In the attention range (0, 0.3), the small attention weight causes many ambiguous nouns to be untranslated, which results in low WSD accuracy.

6.3 Error Distribution

Figure 6 shows the error distributions over absolute frequency (sense frequency in the training set) and relative frequency (sense frequency to source word frequency). The frequency information is given in the test set. It is very clear that most of the errors are in the left bottom corner which are low in both absolute frequency and relative frequency. There are 84.1% and 80.8% errors with an absolute frequency of less than 2000 in RNN and Transformer, respectively.

Even though the attention mechanism pays a lot
of attention to a low-frequency sense, the model is still likely to generate an incorrect translation. Our evaluation method is different from Rios et al. (2017), but the finding is the same, namely that data sparsity leads to incorrect translations.

7 Conclusion

In this paper, we analyze two different attention mechanisms with respect to WSD in NMT. We evaluate the translations of ambiguous nouns directly rather than scoring the contrastive translation pairs, using ContraWSD as the test set. We show that the WSD accuracy of these two models is around 80.2% and 85.5%, respectively. Data sparsity is the main problem causing incorrect translations. We hypothesize that attention mechanisms distribute more attention to context tokens to guide the translation of ambiguous nouns. However, we find that attention mechanisms are likely to pay more attention to the ambiguous noun itself. Compared to vanilla attention mechanisms, we reveal that the first few layers in Transformer attention mechanisms learn to “align” source and target tokens, while the last few layers learn to distribute attention to the related but unaligned context tokens. We conclude that encoder-decoder attention is not the main mechanism used by NMT models to incorporate contextual information for WSD. In addition, Section 6.2 has told us that the larger attention weights, the higher WSD accuracy. Tang et al. (2018b) have shown that Transformer models are better than RNN models in WSD because of their stronger encoding ability. These results suggest that NMT models learn to encode contextual information necessary for WSD in the encoder hidden states.

The question how NMT models learn to represent word senses and similar phenomena has implications for transfer learning, the diagnosis of translation errors, and for the design of architectures for MT, including architectures that scale up the context window to the level of documents. We hope that future work will continue to deepen our understanding of the internal workings of NMT models.

Acknowledgments

We thank all the anonymous reviews who give a lot of valuable and insightful comments. Gongbo Tang is funded by the Chinese Scholarship Council (grant number 201607110016).

References


