Why Self-Attention?
A Targeted Evaluation of Neural Machine Translation Architectures

Gongbo Tang[1*]   **Mathias Müller**[2]   **Annette Rios**[2]   **Rico Sennrich**[2,3]

[1]Department of Linguistics and Philology, Uppsala University

[2]Institute of Computational Linguistics, University of Zurich

[3]School of Informatics, University of Edinburgh

Abstract

Recently, non-recurrent architectures (convolutional, self-attentional) have outperformed RNNs in neural machine translation. CNNs and self-attentional networks can connect distant words via shorter network paths than RNNs, and it has been speculated that this improves their ability to model long-range dependencies. However, this theoretical argument has not been tested empirically, nor have alternative explanations for their strong performance been explored in-depth. We hypothesize that the strong performance of CNNs and self-attentional networks could also be due to their ability to extract semantic features from the source text, and we evaluate RNNs, CNNs and self-attention networks on two tasks: subject-verb agreement (where capturing long-range dependencies is required) and word sense disambiguation (where semantic feature extraction is required). Our experimental results show that: 1) self-attentional networks and CNNs do not outperform RNNs in modeling subject-verb agreement over long distances; 2) self-attentional networks perform distinctly better than RNNs and CNNs on word sense disambiguation.

1 Introduction

Different architectures have been shown to be effective for neural machine translation (NMT), ranging from recurrent architectures (Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2015; Sutskever et al., 2014; Luong et al., 2015) to convolutional (Kalchbrenner and Blunsom, 2013; Gehring et al., 2017) and, most recently, fully self-attentional (Transformer) models (Vaswani et al., 2017). Since comparisons (Gehring et al., 2017; Vaswani et al., 2017; Hieber et al., 2017) are mainly carried out via BLEU (Papineni et al.,

---

2002), it is inherently difficult to attribute gains in BLEU to architectural properties.

Recurrent neural networks (RNNs) (Elman, 1990) can easily deal with variable-length input sentences and thus are a natural choice for the encoder and decoder of NMT systems. Modern variants of RNNs, such as GRUs (Cho et al., 2014) and LSTMs (Hochreiter and Schmidhuber, 1997), address the difficulty of training recurrent networks with long-range dependencies. Gehring et al. (2017) introduce a neural architecture where both the encoder and decoder are based on CNNs, and report better BLEU scores than RNN-based NMT models. Moreover, the computation over all tokens can be fully parallelized during training, which increases efficiency. Vaswani et al. (2017) propose Transformer models, which are built entirely with attention layers, without convolution or recurrence. They report new state-of-art BLEU scores for EN→DE and EN→FR. Yet, the BLEU metric is quite coarse-grained, and offers no insight as to which aspects of translation are improved by different architectures.

To explain the observed improvements in BLEU, previous work has drawn on theoretical arguments. Both Gehring et al. (2017) and Vaswani et al. (2017) argue that the length of the paths in neural networks between co-dependent elements affects the ability to learn these dependencies: the shorter the path, the easier the model learns such dependencies. The papers argue that Transformers and CNNs are better suited than RNNs to capture long-range dependencies.

However, this claim is based on a theoretical argument and has not been empirically tested. We argue other abilities of non-recurrent networks could be responsible for their strong performance. Specifically, we hypothesize that the improvements in BLEU are due to CNNs and Transformers being strong semantic feature extractors.

In this paper, we evaluate all three popular NMT architectures: models based on RNNs (referred to as *RNNS2S* in the remainder of the paper), based on CNNs (referred to as *ConvS2S*) and self-attentional models (referred to as Transformers). Motivated by the aforementioned theoretical claims regarding path length and semantic feature extraction, we evaluate their performance on a subject-verb agreement task (that requires modeling long-range dependencies) and a word sense disambiguation (WSD) task (that requires extracting semantic features). Both tasks build on test sets of contrastive translation pairs, *Lingeval97* (Sennrich, 2017) and *ContraWSD* (Rios et al., 2017).

The main contributions of this paper can be summarized as follows:

- We test the theoretical claims that architectures with shorter paths through networks are better at capturing long-range dependencies. Our experimental results on modeling subject-verb agreement over long distances do not show any evidence that Transformers or CNNs are superior to RNNs in this regard.

- We empirically show that the number of attention heads in Transformers impacts their ability to capture long-distance dependencies. Specifically, many-headed multi-head attention is essential for modeling long-distance phenomena with only self-attention.

- We empirically show that Transformers excel at WSD, indicating that they are strong semantic feature extractors.

2 Related work

Yin et al. (2017) are the first to compare CNNs, LSTMs and GRUs on several NLP tasks. They find that CNNs are better at tasks related to semantics, while RNNs are better at syntax-related tasks, especially for longer sentences.

Based on the work of Linzen et al. (2016), Bernardy and Lappin (2017) find that RNNs perform better than CNNs on a subject-verb agreement task, which is a good proxy for how well long-range dependencies are captured. Tran et al. (2018) find that a Transformer language model performs worse than an RNN language model on a subject-verb agreement task. They, too, note that this is especially true as the distance between subject and verb grows, even if RNNs resulted in a higher perplexity on the validation set. This result of Tran et al. (2018) is clearly in contrast to the general finding that Transformers are better than RNNs for NMT tasks.

Bai et al. (2018) evaluate CNNs and LSTMs on several sequence modeling tasks. They conclude that CNNs are better than RNNs for sequence modeling. However, their CNN models perform much worse than the state-of-art LSTM models on some sequence modeling tasks, as they themselves state in the appendix.

Tang et al. (2018) evaluate different RNN architectures and Transformer models on the task of historical spelling normalization which translates a historical spelling into its modern form. They find that Transformer models surpass RNN models only in high-resource conditions.

In contrast to previous studies, we focus on the machine translation task, where architecture comparisons so far are mostly based on BLEU.

3 Background

3.1 NMT Architectures

We evaluate three different NMT architectures: RNN-based models, CNN-based models, and Transformer-based models. All of them have a bipartite structure in the sense that they consist of an encoder and a decoder. The encoder and the decoder interact via a soft-attention mechanism (Bahdanau et al., 2015; Luong et al., 2015), with one or multiple attention layers.

In the following sections, h_i^l is the hidden state at step i of layer l, h_{i-1}^l represents the hidden state at the previous step of layer l, h_i^{l-1} means the hidden state at i of $l-1$ layer, E_{x_i} represents the embedding of x_i, and $e_{pos,i}$ denotes the positional embedding at position i.

3.1.1 RNN-based NMT

RNNs are stateful networks that change as new inputs are fed to them, and each state has a direct connection only to the previous state. Thus, the path length of any two tokens with a distance of n in RNNs is exactly n. Figure 1 (a) shows an illustration of RNNs.

$$h_i^l = h_i^{l-1} + f_{rnn}(h_i^{l-1}, h_{i-1}^l) \qquad (1)$$

In deep architectures, two adjacent layers are commonly connected with residual connections. In the lth encoder layer, h_i^l is generated by Equation 1,
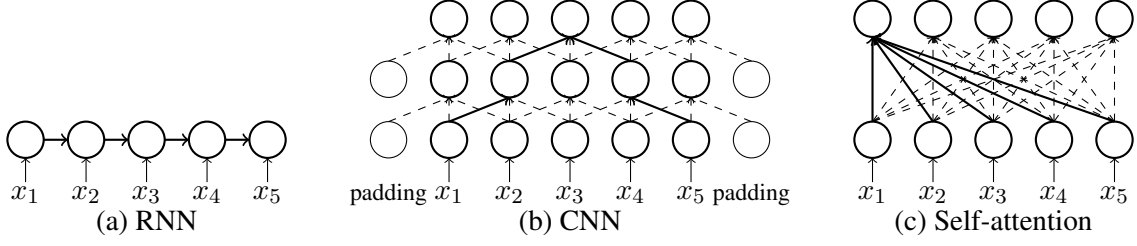
Figure 1: Architectures of different neural networks in NMT.

where f_{rnn} is the RNN (GRU or LSTM) function. In the first layer, $h_i^0 = f_{rnn}(E_{x_i}, h_{i-1}^0)$.

In addition to the connection between the encoder and decoder via attention, the initial state of the decoder is usually initialized with the average of the hidden states or the last hidden state of the encoder.

3.1.2 CNN-based NMT

CNNs are hierarchical networks, in that convolution layers capture local correlations. The local context size depends on the size of the kernel and the number of layers. In order to keep the output the same length as the input, CNN models add padding symbols to input sequences. Given an L-layer CNN with a kernel size k, the largest context size is $L(k-1)$. For any two tokens in a local context with a distance of n, the path between them is only $\lceil n/(k-1) \rceil$.

As Figure 1 (b) shows, a 2-layer CNN with kernel size 3 "sees" an effective local context of 5 tokens. The path between the first token and the fifth token is only 2 convolutions.[1] Since CNNs do not have a means to infer the position of elements in a sequence, positional embeddings are introduced.

$$h_i^l = h_i^{l-1} + f_{cnn}(W^l[h_{i-\lfloor k/2 \rfloor}^{l-1}; ...; h_{i+\lfloor k/2 \rfloor}^{l-1}] + b^l) \quad (2)$$

The hidden state h_i^l shown in Equation 2 is related to the hidden states in the same convolution and the hidden state h_i^{l-1} from the previous layer. k denotes the kernel size of CNNs and f_{cnn} is a non-linearity. *ConvS2S* chooses Gated Linear Units (GLU) which can be viewed as a gated variation of ReLUs. W^l are called convolutional filters. In the input layer, $h_i^0 = E_{x_i} + e_{pos,i}$.

---

[1] Note that the decoder employs masking to avoid conditioning the model on future information, which reduces the effective context size to $L\frac{k-1}{2}$.

3.1.3 Transformer-based NMT

Transformers rely heavily on self-attention networks. Each token is connected to any other token in the same sentence directly via self-attention. Moreover, Transformers feature attention networks with multiple *attention heads*. Multi-head attention is more fine-grained, compared to conventional 1-head attention mechanisms. Figure 1 (c) illustrates that any two tokens are connected directly: the path length between the first and the fifth tokens is 1. Similar to CNNs, positional information is also preserved in positional embeddings.

The hidden state in the Transformer encoder is calculated from all hidden states of the previous layer. The hidden state h_i^l in a self-attention network is computed as in Equation 3.

$$h_i^l = h_i^{l-1} + f(\text{self-attention}(h_i^{l-1})) \quad (3)$$

where f represents a feedforward network with ReLU as the activation function and layer normalization. In the input layer, $h_i^0 = E_{x_i} + e_{pos,i}$. The decoder additionally has a multi-head attention over the encoder hidden states.

3.2 Contrastive Evaluation of Machine Translation

Since we evaluate different NMT architectures explicitly on subject-verb agreement and WSD (both happen implicitly during machine translation), BLEU as a measure of overall translation quality is not helpful. In order to conduct these targeted evaluations, we use contrastive test sets.

Sets of contrastive translations can be used to analyze specific types of errors. Human reference translations are paired with one or more contrastive variants, where a specific type of error is introduced automatically.

The evaluation procedure then exploits the fact that NMT models are conditional language models. By virtue of this, given any source sentence S

and target sentence T, any NMT model can assign to them a probability $P(T|S)$. If a model assigns a higher score to the correct target sentence than to a contrastive variant that contains an error, we consider it a correct decision. The accuracy of a model on such a test set is simply the percentage of cases where the correct target sentence is scored higher than all contrastive variants.

Contrastive evaluation tests the sensitivity of NMT models to specific translation errors. The contrastive examples are designed to capture specific translation errors rather than evaluating the global quality of NMT models. Although they do not replace metrics such as BLEU, they give further insights into the performance of models, on specific linguistic phenomena.

3.2.1 *Lingeval97*

Lingeval97 has over 97,000 English→German contrastive translation pairs featuring different linguistic phenomena, including subject-verb agreement, noun phrase agreement, separable verb-particle constructions, transliterations and polarity. In this paper, we are interested in evaluating the performance on long-range dependencies. Thus, we focus on the subject-verb agreement category which consists of 35,105 instances.

In German, verbs must agree with their subjects in both grammatical number and person. Therefore, in a contrastive translation, the grammatical number of a verb is swapped. Table 1 gives an example.

English:	[...] plan will be approved
German:	[...] **Plan** verabschiedet **wird**
Contrast:	[...] **Plan** verabschiedet **werden**

Table 1: An example of a contrastive pair in the subject-verb agreement category.

3.2.2 *ContraWSD*

In *ContraWSD*, given an ambiguous word in the source sentence, the correct translation is replaced by another meaning of the ambiguous word which is incorrect. For example, in a case where the English word *line* is the correct translation of the German source word *Schlange*, *ContraWSD* replaces *line* with the other translations of *Schlange*, such as *snake*, *serpent*, to generate contrastive translations.

For German→English, *ContraWSD* contains 84 different German word senses. It has 7,200

German→English lexical ambiguities, each lexical ambiguity instance has 3.5 contrastive translations on average. For German→French, it consists of 71 different German word senses. There are 6,700 German→French lexical ambiguities, with an average of 2.2 contrastive translations each lexical ambiguity instance. All the ambiguous words are nouns so that the disambiguation is not possible simply based on syntactic context.

4 Subject-verb Agreement

The subject-verb agreement task is the most popular choice for evaluating the ability to capture long-range dependencies and has been used in many studies (Linzen et al., 2016; Bernardy and Lappin, 2017; Sennrich, 2017; Tran et al., 2018). Thus, we also use this task to evaluate different NMT architectures on long-range dependencies.

4.1 Experimental Settings

Different architectures are hard to compare fairly because many factors affect performance. We aim to create a level playing field for the comparison by training with the same toolkit, *Sockeye* (Hieber et al., 2017) which is based on MXNet (Chen et al., 2015). In addition, different hyperparameters and training techniques (such as label smoothing or layer normalization) have been found to affect the performance (Chen et al., 2018). We apply the same hyperparameters and techniques for all architectures except the parameters of each specific architecture. Since the best hyperparameters for different architectures may be diverse, we verify our hyperparameter choice by comparing our results to those published previously. Our models achieve similar performance to that reported by Hieber et al. (2017) with the best available settings. In addition, we extend *Sockeye* with an interface that enables scoring of existing translations, which is required for contrastive evaluation.

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 and CNNs is 0.2. The kernel size of CNNs is 3. Transformers have an 8-head attention mechanism.

To test the robustness of our findings, we also test a different style of RNN architecture, from a different toolkit. We evaluate bi-deep transitional RNNs (Miceli Barone et al., 2017) which are state-of-art RNNs in machine translation. We use the bi-deep RNN-based model (*RNN-bideep*) implemented in *Marian* (Junczys-Dowmunt et al., 2018). Different from the previous settings, we use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$. The initial learning rate is 0.0003. We tie target embeddings and output embeddings. Both the encoder and decoder have 4 layers of LSTM units, only the encoder layers are bi-directional. LSTM units consist of several cells (deep transition): 4 in the first layer of the decoder, 2 cells everywhere else.

We use training data from the WMT17 shared task.[2] We use *newstest2013* as the validation set, and use *newstest2014* and *newstest2017* as the test sets. All BLEU scores are computed with *Sacre-BLEU* (Post, 2018). 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). We employ the model that has the best perplexity on the validation set for the evaluation.

4.2 Overall Results

Table 2 reports the BLEU scores on *newstest2014* and *newstest2017*, the perplexity on the validation set, and the accuracy on long-range dependencies.[3] *Transformer* achieves the highest accuracy on this task and the highest BLEU scores on both *newstest2014* and *newstest2017*. Compared to *RNNS2S*, *ConvS2S* has slightly better results regarding BLEU scores, but a much lower accuracy on long-range dependencies. The *RNN-bideep* model achieves distinctly better BLEU scores and a higher accuracy on long-range dependencies.

However, it still cannot outperform Transformers on any of the tasks.

Model	2014	2017	PPL	Acc(%)
RNNS2S	23.3	25.1	6.1	95.1
ConvS2S	23.9	25.2	7.0	84.9
Transformer	**26.7**	**27.5**	**4.5**	**97.1**
RNN-bideep	24.7	26.1	5.7	96.3

Table 2: The results of different NMT models, including the BLEU scores on *newstest2014* and *newstest2017*, the perplexity on the validation set, and the accuracy of long-range dependencies.

Figure 2 shows the performance of different architectures on the subject-verb agreement task. It is evident that *Transformer*, *RNNS2S*, and *RNN-bideep* perform much better than *ConvS2S* on long-range dependencies. However, *Transformer*, *RNNS2S*, and *RNN-bideep* are all robust over long distances. *Transformer* outperforms *RNN-bideep* for distances 11-12, but *RNN-bideep* performs equally or better for distance 13 or higher. Thus, we cannot conclude that Transformer models are particularly stronger than RNN models for long distances, despite achieving higher average accuracy on distances above 10.
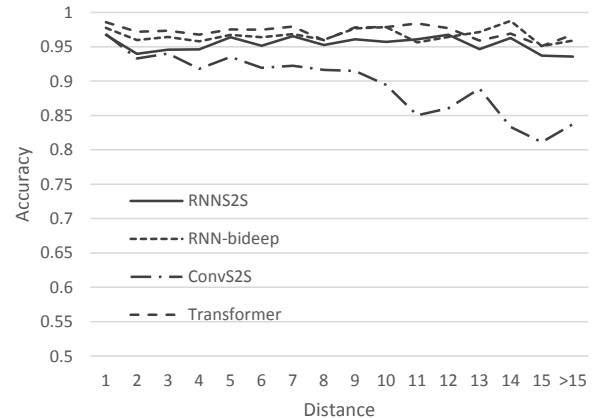

Figure 2: Accuracy of different NMT models on the subject-verb agreement task.

4.2.1 CNNs

Theoretically, the performance of CNNs will drop when the distance between the subject and the verb exceeds the local context size. However, *ConvS2S* is also clearly worse than *RNNS2S* for subject-verb agreement within the local context size.

In order to explore how the ability of *ConvS2S* to capture long-range dependencies depends on the local context size, we train additional systems,

varying the number of layers and kernel size. Table 3 shows the performance of different *ConvS2S* models. Figure 3 displays the performance of two 8-layer CNNs with kernel size 3 and 7, a 6-layer CNN with kernel size 3, and *RNNS2S*. The results indicate that the accuracy increases when the local context size becomes larger, but the BLEU score does not. Moreover, *ConvS2S* is still not as good as *RNNS2S* for subject-verb agreement.

Layer	K	Ctx	2014	2017	Acc(%)
4	3	4	22.9	24.2	81.1
6	3	6	23.6	25.0	82.5
8	3	8	**23.9**	**25.2**	84.9
8	5	16	23.5	24.7	89.7
8	7	24	23.3	24.6	**91.3**

Table 3: The performance of *ConvS2S* with different settings. *K* means the kernel size. The *ctx* column is the theoretical largest local context size in the masked decoder.
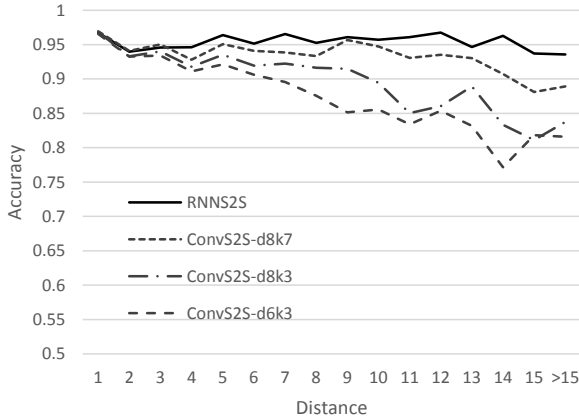

Figure 3: Results of *ConvS2S* models and the *RNNS2S* model at different distances.

Regarding the explanation for the poor performance of *ConvS2S*, we identify the limited context size as a major problem. One assumption to explain the remaining difference is that, scale invariance of CNNs is relatively poor (Xu et al., 2014). Scale-invariance is important in NLP, where the distance between arguments is flexible, and current recurrent or attentional architectures are better suited to handle this variance.

Our empirical results do not confirm the theoretical arguments in Gehring et al. (2017) that CNNs can capture long-range dependencies better with a shorter path. The BLEU score does not correlate well with the targeted evaluation of long-range distance interactions. This is due to the locality

of BLEU, which only measures on the level of n-grams, but it may also indicate that there are other trade-offs between the modeling of different phenomena depending on hyperparameters. If we aim to get better performance on long-range dependencies, we can take this into account when optimizing hyperparameters.

4.2.2 RNNs vs. *Transformer*

Even though *Transformer* achieves much better BLEU scores than *RNNS2S* and *RNN-bideep*, the accuracies of these architectures on long-range dependencies are close to each other in Figure 2.

Our experimental result contrasts with the result from Tran et al. (2018). They find that Transformers perform worse than LSTMs on the subject-verb agreement task, especially when the distance between the subject and the verb becomes longer. We perform several experiments to analyze this discrepancy with Tran et al. (2018).

A first hypothesis is that this is caused by the amount of training data, since we used much larger datasets than Tran et al. (2018). We retrain all the models with a small amount of training data similar to the amount used by Tran et al. (2018), about 135K sentence pairs. The other training settings are the same as in Section 4.1. We do not see the expected degradation of *Transformer-s*, compared to *RNNS2S-s* (see Figure 4). In Table 4, the performance of *RNNS2S-s* and *Transformer-s* is similar, including the BLEU scores on *newstest2014, newstest2017*, the perplexity on the validation set, and the accuracy on the long-range dependencies.
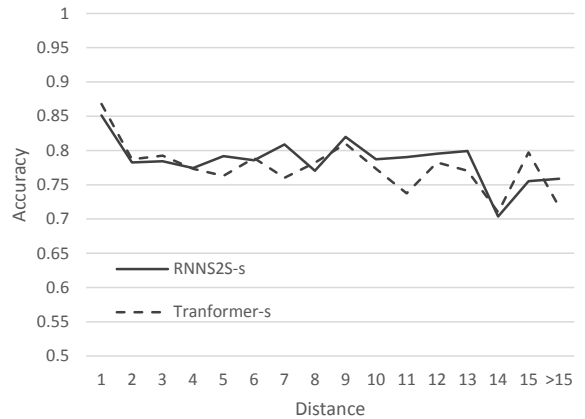

Figure 4: Results of a Transformer and *RNNS2S* model trained on a small dataset.

A second hypothesis is that the experimental settings lead to the different results. In order to investigate this, we do not only use a small training set,

but also replicate the experimental settings of Tran et al. (2018). The main changes are neural network layers (8→4); embedding size (512→128); multi-head size (8→2); dropout rate (0.1→0.2); checkpoint save frequency (4,000→1,000), and initial learning rate (0.0002→0.001).

Model	2014	2017	PPL	Acc(%)
RNNS2S-s	7.3	7.8	47.8	77.3
Trans-s	7.2	8.0	44.6	74.6
RNNS2S-re	9.2	10.5	39.2	77.7
Trans-re-h2	9.6	10.7	36.9	71.9
Trans-re-h4	9.5	11.9	35.8	73.8
Trans-re-h8	9.4	10.4	36.0	75.3

Table 4: The results of different models with small training data and replicate settings. *Trans* is short for Transformer. Models with the suffix "*-s*" are models trained with small data set. Models with the suffix "*-re*" are models trained with replicate settings. "*h2, h4, h8*" indicates the number of attention heads for Transformer models.
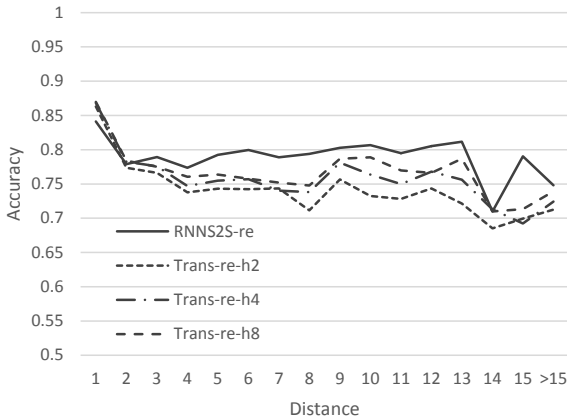

Figure 5: Results of the models with replicate settings, varying the number of attention heads for the Transformer models.

In the end, we get a result that is similar to Tran et al. (2018). In Figure 5, *Transformer-re-h2* performs clearly worse than *RNNS2S-re* on long-range dependencies. By increasing the number of heads in multi-head attention, subject-verb accuracy over long distances can be improved substantially, even though it remains below that of *RNNS2S-re*. Also, the effect on BLEU is small.

Our results suggest that the importance of multi-head attention with a large number of heads is larger than BLEU would suggest, especially for the modeling of long-distance phenomena, since multi-head attention provides a way for the model to attend to both local and distant context, whereas distant context may be overshadowed by local context in an attention mechanism with a single or few heads.

Although our study is not a replication of Tran et al. (2018), who work on a different task and a different test set, our results do suggest an alternative interpretation of their findings, namely that the poor performance of the Transformer in their experiments is due to hyperparameter choice. Rather than concluding that RNNs are superior to Transformers for the modeling of long-range dependency phenomena, we find that the number of heads in multi-head attention affects the ability of Transformers to model long-range dependencies in subject-verb agreement.

5 WSD

Our experimental results on the subject-verb agreement task demonstrate that CNNs and Transformer are not better at capturing long-range dependencies compared to RNNs, even though the paths in CNNs and Transformers are shorter. This finding is not in accord with the theoretical argument in both Gehring et al. (2017) and Vaswani et al. (2017). However, these architectures perform well empirically according to BLEU. Thus, we further evaluate these architectures on WSD, to test our hypothesis that non-recurrent architectures are better at extracting semantic features.

5.1 Experimental settings

We evaluate all architectures on *ContraWSD* on both DE→EN and DE→FR. We reuse the parameter settings in Section 4.1, except that: the initial learning rate of *ConvS2S* is reduced from 0.0003 to 0.0002 in DE→EN; the checkpoint saving frequency is changed from 4,000 to 1,000 in DE→FR because of the training data size.

For DE→EN, the training set, validation set, and test set are as same as the other direction EN→DE. For DE→FR, we use around 2.1 million sentence pairs from Europarl (v7) (Tiedemann, 2012)[4] and News Commentary (v11) cleaned by Rios et al. (2017)[5] as our training set. We use *newstest2013* as the evaluation set, and use *newstest2012* as the test set. All the data is preprocessed with Moses scripts.

---

[4] http://opus.nlpl.eu/Europarl.php
[5] http://data.statmt.org/ContraWSD/

Model	DE→EN				DE→FR		
	PPL	2014	2017	Acc(%)	PPL	2012	Acc(%)
RNNS2S	5.7	29.1	30.1	84.0	7.06	16.4	72.2
ConvS2S	6.3	29.1	30.4	82.3	7.93	16.8	72.7
Transformer	**4.3**	**32.7**	33.7	**90.3**	**4.9**	**18.7**	**76.7**
uedin-wmt17	–	–	**35.1**	87.9	–	–	–
TransRNN	5.2	30.5	31.9	86.1	6.3	17.6	74.2

Table 5: The results of different architectures on *newstest* sets and *ContraWSD*. *PPL* is the perplexity on the validation set. *Acc* means accuracy on the test set.

In addition, we also compare to the best result reported for DE→EN, achieved by *uedin-wmt17* (Sennrich et al., 2017), which is an ensemble of 4 different models and reranked with right-to-left models.[6] *uedin-wmt17* is based on the bi-deep RNNs (Miceli Barone et al., 2017) that we mentioned before. To the original 5.9 million sentence pairs in the training set, they add 10 million synthetic pairs with back-translation.

5.2 Overall Results

Table 5 gives the performance of all the architectures, including the perplexity on validation sets, the BLEU scores on *newstest*, and the accuracy on *ContraWSD*. Transformers distinctly outperform *RNNS2S* and *ConvS2S* models on DE→EN and DE→FR. Moreover, the Transformer model on DE→EN also achieves higher accuracy than *uedin-wmt17*, although the BLEU score on *newstest2017* is 1.4 lower than *uedin-wmt17*. We attribute this discrepancy between BLEU and WSD performance to the use of synthetic news training data in *uedin-wmt17*, which causes a large boost in BLEU due to better domain adaptation to *newstest*, but which is less helpful for ContraWSD, whose test set is drawn from a variety of domains.

For DE→EN, *RNNS2S* and *ConvS2S* have the same BLEU score on *newstest2014*, *ConvS2S* has a higher score on *newstest2017*. However, the WSD accuracy of *ConvS2S* is 1.7% lower than *RNNS2S*. For DE→FR, *ConvS2S* achieves slightly better results on both BLEU scores and accuracy than *RNNS2S*.

The Transformer model strongly outperforms the other architectures on this WSD task, with a gap of 4–8 percentage points. This affirms our hypothesis that Transformers are strong semantic features extractors.

5.3 Hybrid Encoder-Decoder Model

In recent work, Chen et al. (2018) find that hybrid architectures with a Transformer encoder and an RNN decoder can outperform a pure Transformer model. They speculate that the Transformer encoder is better at encoding or extracting features than the RNN encoder, whereas the RNN is better at conditional language modeling.

For WSD, it is unclear whether the most important component is the encoder, the decoder, or both. Following the hypothesis that Transformer encoders excel as semantic feature extractors, we train a hybrid encoder-decoder model (*TransRNN*) with a Transformer encoder and an RNN decoder.

The results (in Table 5) show that *TransRNN* performs better than *RNNS2S*, but worse than the pure Transformer, both in terms of BLEU and WSD accuracy. This indicates that WSD is not only done in the encoder, but that the decoder also affects WSD performance. We note that Chen et al. (2018); Domhan (2018) introduce the techniques in Transformers into RNN-based models, with reportedly higher BLEU. Thus, it would be interesting to see if the same result holds true with their architectures.

6 Conclusion

In this paper, we evaluate three popular NMT architectures, *RNNS2S*, *ConvS2S*, and Transformers, on subject-verb agreement and WSD by scoring contrastive translation pairs.

We test the theoretical claims that shorter path lengths make models better capture long-range dependencies. Our experimental results show that:

- There is no evidence that CNNs and Transformers, which have shorter paths through networks, are empirically superior to RNNs in modeling subject-verb agreement over long distances.

- The number of heads in multi-head attention affects the ability of a Transformer to model long-range dependencies in the subject-verb agreement task.

- Transformer models excel at another task, WSD, compared to the CNN and RNN architectures we tested.

Lastly, our findings suggest that assessing the performance of NMT architectures means finding their inherent trade-offs, rather than simply computing their overall BLEU score. A clear understanding of those strengths and weaknesses is important to guide further work. Specifically, given the idiosyncratic limitations of recurrent and self-attentional models, combining them is an exciting line of research. The apparent weakness of CNN architectures on long-distance phenomena is also a problem worth tackling, and we can find inspiration from related work in computer vision (Xu et al., 2014).

Acknowledgments

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.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Jean-Philippe Bernardy and Shalom Lappin. 2017. Using Deep Neural Networks to Learn Syntactic Agreement. *LiLT (Linguistic Issues in Language Technology)*, 15(2):1–15.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86. Association for Computational Linguistics.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*.

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.

Tobias Domhan. 2018. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1799–1808. Association for Computational Linguistics.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

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

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*.

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

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. *arXiv preprint arXiv:1804.00344*.

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.

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.

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.

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep architectures for Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pages 99–107, Copenhagen, Denmark. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

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.

Rico Sennrich. 2017. How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's Neural MT Systems for WMT17. In *Proceedings of the Second Conference on Machine Translation*, pages 389–399, Copenhagen, Denmark. 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 *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., Montréal, Canada.

Gongbo Tang, Fabienne Cap, Eva Pettersson, and Joakim Nivre. 2018. An evaluation of neural machine translation models on historical spelling normalization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1320–1331. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ke Tran, Arianna Bisazza, and Christof Monz. 2018. The Importance of Being Recurrent for Modeling Hierarchical Structure. *arXiv preprint arXiv:1803.03585*.

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

Yichong Xu, Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, and Zheng Zhang. 2014. Scale-invariant convolutional neural networks. *arXiv preprint arXiv:1411.6369*.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.