Effective Approaches to Attention-based Neural Machine Translation

Minh-Thang Luong **Hieu Pham** **Christopher D. Manning**
Computer Science Department, Stanford University, Stanford, CA 94305
{lmthang,hyhieu,manning}@stanford.edu

Abstract

An attentional mechanism has lately been used to improve neural machine translation (NMT) by selectively focusing on parts of the source sentence during translation. However, there has been little work exploring useful architectures for attention-based NMT. This paper examines two simple and effective classes of attentional mechanism: a *global* approach which always attends to all source words and a *local* one that only looks at a subset of source words at a time. We demonstrate the effectiveness of both approaches on the WMT translation tasks between English and German in both directions. With local attention, we achieve a significant gain of 5.0 BLEU points over non-attentional systems that already incorporate known techniques such as dropout. Our ensemble model using different attention architectures yields a new state-of-the-art result in the WMT'15 English to German translation task with 25.9 BLEU points, an improvement of 1.0 BLEU points over the existing best system backed by NMT and an n-gram reranker.[1]

1 Introduction

Neural Machine Translation (NMT) achieved state-of-the-art performances in large-scale translation tasks such as from English to French (Luong et al., 2015) and English to German (Jean et al., 2015). NMT is appealing since it requires minimal domain knowledge and is conceptually simple. The model by Luong et al. (2015) reads through all the source words until the end-of-sentence symbol <eos> is reached. It then starts emitting one target word at a time, as illustrated in Figure 1. NMT

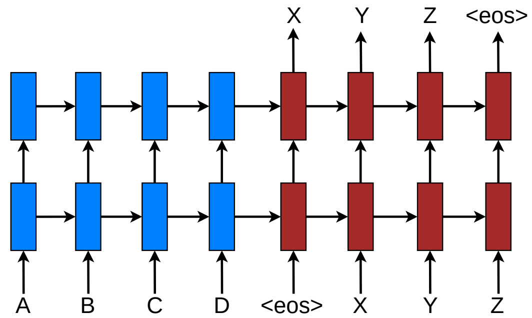

Figure 1: **Neural machine translation** – a stacking recurrent architecture for translating a source sequence `A B C D` into a target sequence `X Y Z`. Here, <eos> marks the end of a sentence.

is often a large neural network that is trained in an end-to-end fashion and has the ability to generalize well to very long word sequences. This means the model does not have to explicitly store gigantic phrase tables and language models as in the case of standard MT; hence, NMT has a small memory footprint. Lastly, implementing NMT decoders is easy unlike the highly intricate decoders in standard MT (Koehn et al., 2003).

In parallel, the concept of "attention" has gained popularity recently in training neural networks, allowing models to learn alignments between different modalities, e.g., between image objects and agent actions in the dynamic control problem (Mnih et al., 2014), between speech frames and text in the speech recognition task (Chorowski et al., 2014), or between visual features of a picture and its text description in the image caption generation task (Xu et al., 2015). In the context of NMT, Bahdanau et al. (2015) has successfully applied such attentional mechanism to jointly translate and align words. To the best of our knowledge, there has not been any other work exploring the use of attention-based architectures for NMT.

In this work, we design, with simplicity and effectiveness in mind, two novel types of attention-

[1] All our code and models are publicly available at http://nlp.stanford.edu/projects/nmt.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1412–1421,
Lisbon, Portugal, 17-21 September 2015.

based models: a *global* approach in which all source words are attended and a *local* one whereby only a subset of source words are considered at a time. The former approach resembles the model of (Bahdanau et al., 2015) but is simpler architecturally. The latter can be viewed as an interesting blend between the *hard* and *soft* attention models proposed in (Xu et al., 2015): it is computationally less expensive than the global model or the soft attention; at the same time, unlike the hard attention, the local attention is differentiable, making it easier to implement and train.[2] Besides, we also examine various alignment functions for our attention-based models.

Experimentally, we demonstrate that both of our approaches are effective in the WMT translation tasks between English and German in both directions. Our attentional models yield a boost of up to 5.0 BLEU over non-attentional systems which already incorporate known techniques such as dropout. For English to German translation, we achieve new state-of-the-art (SOTA) results for both WMT'14 and WMT'15, outperforming previous SOTA systems, backed by NMT models and n-gram LM rerankers, by more than 1.0 BLEU. We conduct extensive analysis to evaluate our models in terms of learning, the ability to handle long sentences, choices of attentional architectures, alignment quality, and translation outputs.

2 Neural Machine Translation

A neural machine translation system is a neural network that directly models the conditional probability $p(y|x)$ of translating a source sentence, $x_1, \ldots, x_n$, to a target sentence, $y_1, \ldots, y_m$.[3] A basic form of NMT consists of two components: (a) an *encoder* which computes a representation $\boldsymbol{s}$ for each source sentence and (b) a *decoder* which generates one target word at a time and hence decomposes the conditional probability as:

$$\log p(y|x) = \sum\nolimits_{j=1}^{m} \log p\left(y_j | y_{<j}, \boldsymbol{s}\right) \quad (1)$$

A natural choice to model such a decomposition in the decoder is to use a recurrent neural network (RNN) architecture, which most of the recent NMT work such as (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Jean et al., 2015) have in common. They, however, differ in terms of which RNN architectures are used for the decoder and how the encoder computes the source sentence representation $\boldsymbol{s}$.

Kalchbrenner and Blunsom (2013) used an RNN with the standard hidden unit for the decoder and a convolutional neural network for encoding the source sentence representation. On the other hand, both Sutskever et al. (2014) and Luong et al. (2015) stacked multiple layers of an RNN with a Long Short-Term Memory (LSTM) hidden unit for both the encoder and the decoder. Cho et al. (2014), Bahdanau et al. (2015), and Jean et al. (2015) all adopted a different version of the RNN with an LSTM-inspired hidden unit, the gated recurrent unit (GRU), for both components.[4]

In more detail, one can parameterize the probability of decoding each word y_j as:

$$p\left(y_j | y_{<j}, \boldsymbol{s}\right) = \mathrm{softmax}\left(g\left(\boldsymbol{h}_j\right)\right) \quad (2)$$

with g being the transformation function that outputs a vocabulary-sized vector.[5] Here, $\boldsymbol{h}_j$ is the RNN hidden unit, abstractly computed as:

$$\boldsymbol{h}_j = f(\boldsymbol{h}_{j-1}, \boldsymbol{s}), \quad (3)$$

where f computes the current hidden state given the previous hidden state and can be either a vanilla RNN unit, a GRU, or an LSTM unit. In (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014; Luong et al., 2015), the source representation $\boldsymbol{s}$ is only used once to initialize the decoder hidden state. On the other hand, in (Bahdanau et al., 2015; Jean et al., 2015) and this work, $\boldsymbol{s}$, in fact, implies a set of source hidden states which are consulted throughout the entire course of the translation process. Such an approach is referred to as an attention mechanism, which we will discuss next.

In this work, following (Sutskever et al., 2014; Luong et al., 2015), we use the stacking LSTM architecture for our NMT systems, as illustrated in Figure 1. We use the LSTM unit defined in (Zaremba et al., 2015). Our training objective is formulated as follows:

$$J_t = \sum\nolimits_{(x,y)\in D} -\log p(y|x) \quad (4)$$

[2] There is a recent work by Gregor et al. (2015), which is very similar to our local attention and applied to the image generation task. However, as we detail later, our model is much simpler and can achieve good performance for NMT.

[3] All sentences are assumed to terminate with a special "end-of-sentence" token <eos>.

[4] They all used a single RNN layer except for the latter two works which utilized a bidirectional RNN for the encoder.

[5] One can provide g with other inputs such as the currently predicted word y_j as in (Bahdanau et al., 2015).

with D being our parallel training corpus.

3 Attention-based Models

Our various attention-based models are classifed into two broad categories, *global* and *local*. These classes differ in terms of whether the "attention" is placed on all source positions or on only a few source positions. We illustrate these two model types in Figure 2 and 3 respectively.

Common to these two types of models is the fact that at each time step t in the decoding phase, both approaches first take as input the hidden state $\boldsymbol{h}_t$ at the top layer of a stacking LSTM. The goal is then to derive a context vector $\boldsymbol{c}_t$ that captures relevant source-side information to help predict the current target word y_t. While these models differ in how the context vector $\boldsymbol{c}_t$ is derived, they share the same subsequent steps.

Specifically, given the target hidden state $\boldsymbol{h}_t$ and the source-side context vector $\boldsymbol{c}_t$, we employ a simple concatenation layer to combine the information from both vectors to produce an attentional hidden state as follows:

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W_c}[\boldsymbol{c}_t;\boldsymbol{h}_t]) \tag{5}$$

The attentional vector $\tilde{\boldsymbol{h}}_t$ is then fed through the softmax layer to produce the predictive distribution formulated as:

$$p(y_t|y_{<t},x) = \mathrm{softmax}(\boldsymbol{W_s}\tilde{\boldsymbol{h}}_t) \tag{6}$$

We now detail how each model type computes the source-side context vector $\boldsymbol{c}_t$.

3.1 Global Attention

The idea of a global attentional model is to consider all the hidden states of the encoder when deriving the context vector c_t. In this model type, a variable-length alignment vector $\boldsymbol{a}_t$, whose size equals the number of time steps on the source side, is derived by comparing the current target hidden state $\boldsymbol{h}_t$ with each source hidden state $\bar{\boldsymbol{h}}_s$:

$$\begin{aligned}\boldsymbol{a}_t(s) &= \mathrm{align}(\boldsymbol{h}_t,\bar{\boldsymbol{h}}_s) \\ &= \frac{\exp\left(\mathrm{score}(\boldsymbol{h}_t,\bar{\boldsymbol{h}}_s)\right)}{\sum_{s'}\exp\left(\mathrm{score}(\boldsymbol{h}_t,\bar{\boldsymbol{h}}_{s'})\right)}\end{aligned} \tag{7}$$

Here, score is referred as a *content-based* function for which we consider three different alternatives:

$$\mathrm{score}(\boldsymbol{h}_t,\bar{\boldsymbol{h}}_s) = \begin{cases}\boldsymbol{h}_t^\top\bar{\boldsymbol{h}}_s & \textit{dot}\\ \boldsymbol{h}_t^\top\boldsymbol{W_a}\bar{\boldsymbol{h}}_s & \textit{general}\\ \boldsymbol{W_a}[\boldsymbol{h}_t;\bar{\boldsymbol{h}}_s] & \textit{concat}\end{cases} \tag{8}$$

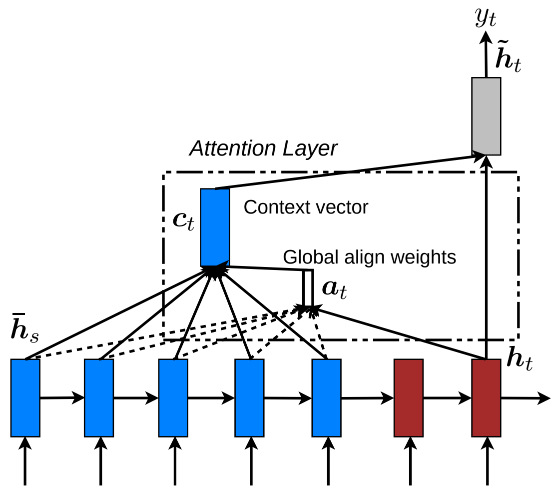

Figure 2: **Global attentional model** – at each time step t, the model infers a *variable-length* alignment weight vector $\boldsymbol{a}_t$ based on the current target state $\boldsymbol{h}_t$ and all source states $\bar{\boldsymbol{h}}_s$. A global context vector $\boldsymbol{c}_t$ is then computed as the weighted average, according to $\boldsymbol{a}_t$, over all the source states.

Besides, in our early attempts to build attention-based models, we use a *location-based* function in which the alignment scores are computed from solely the target hidden state $\boldsymbol{h}_t$ as follows:

$$\boldsymbol{a}_t = \mathrm{softmax}(\boldsymbol{W_a}\boldsymbol{h}_t) \qquad \textit{location} \tag{9}$$

Given the alignment vector as weights, the context vector c_t is computed as the weighted average over all the source hidden states.[6]

Comparison to (Bahdanau et al., 2015) – While our global attention approach is similar in spirit to the model proposed by Bahdanau et al. (2015), there are several key differences which reflect how we have both simplified and generalized from the original model. First, we simply use hidden states at the top LSTM layers in both the encoder and decoder as illustrated in Figure 2. Bahdanau et al. (2015), on the other hand, use the concatenation of the forward and backward source hidden states in the bi-directional encoder and target hidden states in their non-stacking uni-directional decoder. Second, our computation path is simpler; we go from $\boldsymbol{h}_t \rightarrow \boldsymbol{a}_t \rightarrow \boldsymbol{c}_t \rightarrow \tilde{\boldsymbol{h}}_t$ then make a prediction as detailed in Eq. (5), Eq. (6), and Figure 2. On the other hand, at any time t, Bahdanau et al. (2015) build from the previous hidden state $\boldsymbol{h}_{t-1} \rightarrow \boldsymbol{a}_t \rightarrow \boldsymbol{c}_t \rightarrow \boldsymbol{h}_t$, which, in turn,

[6] Eq. (9) implies that all alignment vectors $\boldsymbol{a}_t$ are of the same length. For short sentences, we only use the top part of $\boldsymbol{a}_t$ and for long sentences, we ignore words near the end.

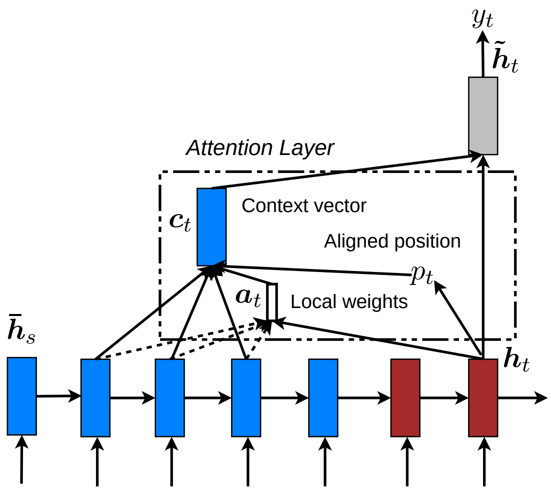

Figure 3: **Local attention model** – the model first predicts a single aligned position p_t for the current target word. A window centered around the source position p_t is then used to compute a context vector $\boldsymbol{c}_t$, a weighted average of the source hidden states in the window. The weights $\boldsymbol{a}_t$ are inferred from the current target state $\boldsymbol{h}_t$ and those source states $\bar{\boldsymbol{h}}_s$ in the window.

goes through a deep-output and a maxout layer before making predictions.[7] Lastly, Bahdanau et al. (2015) only experimented with one alignment function, the *concat* product; whereas we show later that the other alternatives are better.

3.2 Local Attention

The global attention has a drawback that it has to attend to all words on the source side for each target word, which is expensive and can potentially render it impractical to translate longer sequences, e.g., paragraphs or documents. To address this deficiency, we propose a *local* attentional mechanism that chooses to focus only on a small subset of the source positions per target word.

This model takes inspiration from the tradeoff between the *soft* and *hard* attentional models proposed by Xu et al. (2015) to tackle the image caption generation task. In their work, soft attention refers to the global attention approach in which weights are placed "softly" over all patches in the source image. The hard attention, on the other hand, selects one patch of the image to attend to at a time. While less expensive at inference time, the hard attention model is non-differentiable and requires more complicated techniques such as variance reduction or reinforcement learning to train.

[7]We will refer to this difference again in Section 3.3.

Our local attention mechanism selectively focuses on a small window of context and is differentiable. This approach has an advantage of avoiding the expensive computation incurred in the soft attention and at the same time, is easier to train than the hard attention approach. In concrete details, the model first generates an aligned position p_t for each target word at time t. The context vector $\boldsymbol{c}_t$ is then derived as a weighted average over the set of source hidden states within the window $[p_t-D, p_t+D]$; D is empirically selected.[8] Unlike the global approach, the local alignment vector $\boldsymbol{a}_t$ is now fixed-dimensional, i.e., $\in \mathbb{R}^{2D+1}$. We consider two variants of the model as below.

Monotonic alignment (**local-m**) – we simply set $p_t = t$ assuming that source and target sequences are roughly monotonically aligned. The alignment vector $\boldsymbol{a}_t$ is defined according to Eq. (7).[9]

Predictive alignment (**local-p**) – instead of assuming monotonic alignments, our model predicts an aligned position as follows:

$$p_t = S \cdot \text{sigmoid}(\boldsymbol{v}_p^\top \tanh(\boldsymbol{W_p h}_t)), \quad (10)$$

$\boldsymbol{W_p}$ and $\boldsymbol{v}_p$ are the model parameters which will be learned to predict positions. S is the source sentence length. As a result of sigmoid, $p_t \in [0, S]$. To favor alignment points near p_t, we place a Gaussian distribution centered around p_t . Specifically, our alignment weights are now defined as:

$$\boldsymbol{a}_t(s) = \text{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) \exp\left(-\frac{(s-p_t)^2}{2\sigma^2}\right) \quad (11)$$

We use the same align function as in Eq. (7) and the standard deviation is empirically set as $\sigma = \frac{D}{2}$. It is important to note that p_t is a *real* nummber; whereas s is an *integer* within the window centered at p_t.[10]

Comparison to (Gregor et al., 2015) – have proposed a *selective attention* mechanism, very similar to our local attention, for the image generation task. Their approach allows the model to select an image patch of varying location and zoom. We, instead, use the same "zoom" for all target positions, which greatly simplifies the formulation and still achieves good performance.

[8]If the window crosses the sentence boundaries, we simply ignore the outside part and consider words in the window.

[9]*local-m* is the same as the global model except that the vector $\boldsymbol{a}_t$ is fixed-length and shorter.

[10]*local-p* is similar to the local-m model except that we dynamically compute p_t and use a Gaussian distribution to modify the original alignment weights $\text{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)$ as shown in Eq. (11). By utilizing p_t to derive $\boldsymbol{a}_t$, we can compute backprop gradients for $\boldsymbol{W_p}$ and $\boldsymbol{v}_p$.

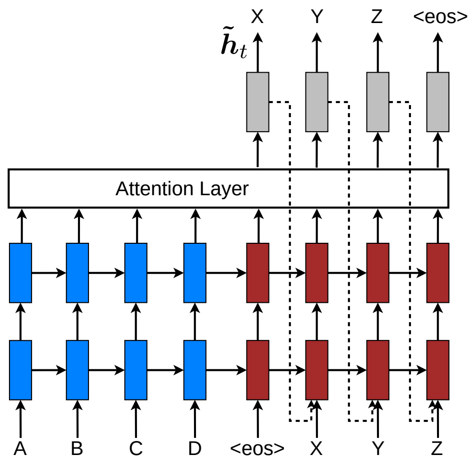

Figure 4: **Input-feeding approach** – Attentional vectors $\tilde{\boldsymbol{h}}_t$ are fed as inputs to the next time steps to inform the model about past alignment decisions.

3.3 Input-feeding Approach

In our proposed global and local approaches, the attentional decisions are made independently, which is suboptimal. Whereas, in standard MT, a *coverage* set is often maintained during the translation process to keep track of which source words have been translated. Likewise, in attentional NMTs, alignment decisions should be made jointly taking into account past alignment information. To address that, we propose an *input-feeding* approach in which attentional vectors $\tilde{\boldsymbol{h}}_t$ are concatenated with inputs at the next time steps as illustrated in Figure 4.[11] The effects of having such connections are two-fold: (a) we hope to make the model fully aware of previous alignment choices and (b) we create a very deep network spanning both horizontally and vertically.

Comparison to other work – Bahdanau et al. (2015) use context vectors, similar to our $\boldsymbol{c}_t$, in building subsequent hidden states, which can also achieve the "coverage" effect. However, there has not been any analysis of whether such connections are useful as done in this work. Also, our approach is more general; as illustrated in Figure 4, it can be applied to general stacking recurrent architectures, including non-attentional models.

Xu et al. (2015) propose a *doubly attentional* approach with an additional constraint added to the training objective to make sure the model pays equal attention to all parts of the image during the caption generation process. Such a constraint can also be useful to capture the coverage set effect in NMT that we mentioned earlier. However, we chose to use the input-feeding approach since it provides flexibility for the model to decide on any attentional constraints it deems suitable.

4 Experiments

We evaluate the effectiveness of our models on the WMT translation tasks between English and German in both directions. newstest2013 (3000 sentences) is used as a development set to select our hyperparameters. Translation performances are reported in case-sensitive BLEU (Papineni et al., 2002) on newstest2014 (2737 sentences) and newstest2015 (2169 sentences). Following (Luong et al., 2015), we report translation quality using two types of BLEU: (a) *tokenized*[12] BLEU to be comparable with existing NMT work and (b) *NIST*[13] BLEU to be comparable with WMT results.

4.1 Training Details

All our models are trained on the WMT'14 training data consisting of 4.5M sentences pairs (116M English words, 110M German words). Similar to (Jean et al., 2015), we limit our vocabularies to be the top 50K most frequent words for both languages. Words not in these shortlisted vocabularies are converted into a universal token <`unk`>.

When training our NMT systems, following (Bahdanau et al., 2015; Jean et al., 2015), we filter out sentence pairs whose lengths exceed 50 words and shuffle mini-batches as we proceed. Our stacking LSTM models have 4 layers, each with 1000 cells, and 1000-dimensional embeddings. We follow (Sutskever et al., 2014; Luong et al., 2015) in training NMT with similar settings: (a) our parameters are uniformly initialized in $[-0.1, 0.1]$, (b) we train for 10 epochs using plain SGD, (c) a simple learning rate schedule is employed – we start with a learning rate of 1; after 5 epochs, we begin to halve the learning rate every epoch, (d) our mini-batch size is 128, and (e) the normalized gradient is rescaled whenever its norm exceeds 5. Additionally, we also use dropout for our LSTMs as suggested by (Zaremba et al., 2015). For dropout models, we train for 12 epochs and start halving the learning rate after 8 epochs.

Our code is implemented in MATLAB. When

[11] If n is the number of LSTM cells, the input size of the first LSTM layer is $2n$; those of subsequent layers are n.

[12] All texts are tokenized with `tokenizer.perl` and BLEU scores are computed with `multi-bleu.perl`.

[13] With the `mteval-v13a` script as per WMT guideline.

System	Ppl	BLEU
Winning WMT'14 system – *phrase-based + large LM* (Buck et al., 2014)		20.7
Existing NMT systems		
RNNsearch (Jean et al., 2015)		16.5
RNNsearch + unk replace (Jean et al., 2015)		19.0
RNNsearch + unk replace + large vocab + *ensemble* 8 models (Jean et al., 2015)		**21.6**
Our NMT systems		
Base	10.6	11.3
Base + reverse	9.9	12.6 (+*1.3*)
Base + reverse + dropout	8.1	14.0 (+*1.4*)
Base + reverse + dropout + global attention (*location*)	7.3	16.8 (+*2.8*)
Base + reverse + dropout + global attention (*location*) + feed input	6.4	18.1 (+*1.3*)
Base + reverse + dropout + local-p attention (*general*) + feed input	5.9	19.0 (+*0.9*)
Base + reverse + dropout + local-p attention (*general*) + feed input + unk replace		20.9 (+*1.9*)
Ensemble 8 models + unk replace		**23.0** (+*2.1*)

Table 1: **WMT'14 English-German results** – shown are the perplexities (ppl) and the *tokenized* BLEU scores of various systems on newstest2014. We highlight the **best** system in bold and give *progressive* improvements in italic between consecutive systems. *local-p* referes to the local attention with predictive alignments. We indicate for each attention model the alignment score function used in pararentheses.

running on a single GPU device Tesla K40, we achieve a speed of 1K *target* words per second. It takes 7–10 days to completely train a model.

4.2 English-German Results

We compare our NMT systems in the English-German task with various other systems. These include the winning system in WMT'14 (Buck et al., 2014), a phrase-based system whose language models were trained on a huge monolingual text, the Common Crawl corpus. For end-to-end neural machine translation systems, to the best of our knowledge, (Jean et al., 2015) is the only work experimenting with this language pair and currently the SOTA system. We only present results for some of our attention models and will later analyze the rest in Section 5.

As shown in Table 1, we achieve progressive improvements when (a) reversing the source sentence, +1.3 BLEU, as proposed in (Sutskever et al., 2014) and (b) using dropout, +1.4 BLEU. On top of that, (c) the global attention approach gives a significant boost of +2.8 BLEU, making our model slightly better than the base attentional system of Bahdanau et al. (2015) (row *RNNSearch*). When (d) using the *input-feeding* approach, we seize another notable gain of +1.3 BLEU and outperform their system. The local attention model with predictive alignments (row *local-p*) proves to be even better, giving us a further improvement of +0.9 BLEU on top of the global attention model. It is interesting to observe the trend previously reported in (Luong et al., 2015) that perplexity strongly correlates with translation quality. In total, we achieve a significant gain of 5.0 BLEU points over the non-attentional baseline, which already includes known techniques such as source reversing and dropout.

The unknown replacement technique proposed in (Luong et al., 2015; Jean et al., 2015) yields another nice gain of +1.9 BLEU, demonstrating that our attentional models do learn useful alignments for unknown works. Finally, by ensembling 8 different models of various settings, e.g., using different attention approaches, with and without dropout etc., we were able to achieve a *new SOTA* result of 23.0 BLEU, outperforming the existing best system (Jean et al., 2015) by +1.4 BLEU.

System	BLEU
SOTA – *NMT + 5-gram rerank* (MILA)	24.9
Our ensemble 8 models + unk replace	**25.9**

Table 2: **WMT'15 English-German results** – *NIST* BLEU scores of the existing WMT'15 SOTA system and our best one on newstest2015.

Latest results in WMT'15 – despite the fact that our models were trained on WMT'14 with slightly less data, we test them on newstest2015 to demonstrate that they can generalize well to different test sets. As shown in Table 2, our best system es-

System	Ppl.	BLEU
WMT'15 systems		
SOTA – *phrase-based* (Edinburgh)		**29.2**
NMT + 5-gram rerank (MILA)		27.6
Our NMT systems		
Base (reverse)	14.3	16.9
+ global (*location*)	12.7	19.1 (+*2.2*)
+ global (*location*) + feed	10.9	20.1 (+*1.0*)
+ global (*dot*) + drop + feed	9.7	22.8 (+*2.7*)
+ global (*dot*) + drop + feed + unk		24.9 (+*2.1*)

Table 3: **WMT'15 German-English results** – performances of various systems (similar to Table 1). The *base* system already includes source reversing on which we add *global* attention, *drop*out, input *feed*ing, and *unk* replacement.

tablishes a *new SOTA* performance of 25.9 BLEU, outperforming the existing best system backed by NMT and a 5-gram LM reranker by +1.0 BLEU.

4.3 German-English Results

We carry out a similar set of experiments for the WMT'15 translation task from German to English. While our systems have not yet matched the performance of the SOTA system, we nevertheless show the effectiveness of our approaches with large and progressive gains in terms of BLEU as illustrated in Table 3. The *attentional* mechanism gives us +2.2 BLEU gain and on top of that, we obtain another boost of up to +1.0 BLEU from the *input-feeding* approach. Using a better alignment function, the content-based *dot* product one, together with *dropout* yields another gain of +2.7 BLEU. Lastly, when applying the unknown word replacement technique, we seize an additional +2.1 BLEU, demonstrating the usefulness of attention in aligning rare words.

5 Analysis

We conduct extensive analysis to better understand our models in terms of learning, the ability to handle long sentences, choices of attentional architectures, and alignment quality. All models considered here are English-German NMT systems tested on newstest2014.

5.1 Learning curves

We compare models built on top of one another as listed in Table 1. It is pleasant to observe in Figure 5 a clear separation between non-attentional and attentional models. The input-feeding ap-

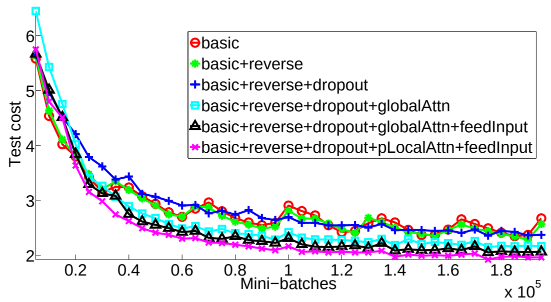

Figure 5: **Learning curves** – test cost ($\ln$ perplexity) on newstest2014 for English-German NMTs as training progresses.

proach and the local attention model also demonstrate their abilities in driving the test costs lower. The non-attentional model with dropout (the blue + curve) learns slower than other non-dropout models, but as time goes by, it becomes more robust in terms of minimizing test errors.

5.2 Effects of Translating Long Sentences

We follow (Bahdanau et al., 2015) to group sentences of similar lengths together and compute a BLEU score per group. As demonstrated in Figure 6, our attentional models are more effective than the other non-attentional model in handling long sentences: the translation quality does not degrade as sentences become longer. Our best model (the blue + curve) outperforms all other systems in all length buckets.

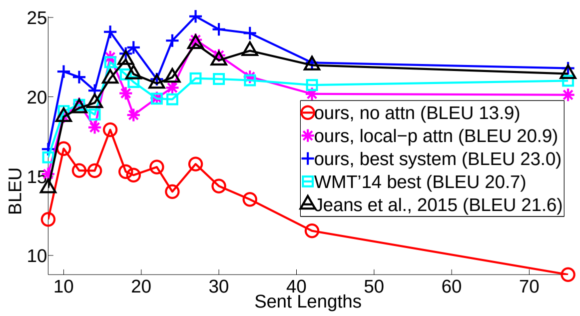

Figure 6: **Length Analysis** – translation qualities of different systems as sentences become longer.

5.3 Choices of Attentional Architectures

We examine different attention models (*global, local-m, local-p*) and different alignment functions (*location, dot, general, concat*) as described in Section 3. Due to limited resources, we cannot run all the possible combinations. However,

System	Ppl	BLEU	
		Before	After unk
global (location)	6.4	18.1	19.3 (+1.2)
global (dot)	6.1	18.6	20.5 (+1.9)
global (general)	6.1	17.3	19.1 (+1.8)
local-m (dot)	>7.0	x	x
local-m (general)	6.2	18.6	20.4 (+1.8)
local-p (dot)	6.6	18.0	19.6 (+1.9)
local-p (general)	**5.9**	**19**	**20.9 (+1.9)**

Table 4: **Attentional Architectures** – performances of different attentional models. We trained two local-m (dot) models; both have ppl > 7.0.

results in Table 4 do give us some idea about different choices. The *location-based* function does not learn good alignments: the *global (location)* model can only obtain a small gain when performing unknown word replacement compared to using other alignment functions.[14] For *content-based* functions, our implementation of *concat* does not yield good performances and more analysis should be done to understand the reason.[15] It is interesting to observe that *dot* works well for the global attention and *general* is better for the local attention. Among the different models, the local attention model with predictive alignments (*local-p*) is best, both in terms of perplexities and BLEU.

5.4 Alignment Quality

A by-product of attentional models are word alignments. While (Bahdanau et al., 2015) visualized alignments for some sample sentences and observed gains in translation quality as an indication of a working attention model, no work has assessed the alignments learned as a whole. In contrast, we set out to evaluate the alignment quality using the alignment error rate (AER) metric.

Given the gold alignment data provided by RWTH for 508 English-German Europarl sentences, we "force" decode our attentional models to produce translations that match the references. We extract only one-to-one alignments by selecting the source word with the highest alignment weight per target word. Nevertheless, as shown in Table 6, we were able to achieve AER scores comparable to the one-to-many alignments obtained by the Berkeley aligner (Liang et al., 2006).[16]

Method	AER
global (location)	0.39
local-m (general)	0.34
local-p (general)	0.36
ensemble	0.34
Berkeley Aligner	0.32

Table 6: **AER scores** – results of various models on the RWTH English-German alignment data.

We also found that the alignments produced by local attention models achieve lower AERs than those of the global one. The AER obtained by the ensemble, while good, is not better than the local-m AER, suggesting the well-known observation that AER and translation scores are not well correlated (Fraser and Marcu, 2007). Due to space constraint, we can only show alignment visualizations in the arXiv version of our paper.[17]

5.5 Sample Translations

We show in Table 5 sample translations in both directions. It it appealing to observe the effect of attentional models in correctly translating names such as "Miranda Kerr" and "Roger Dow". Non-attentional models, while producing sensible names from a language model perspective, lack the direct connections from the source side to make correct translations.

We also observed an interesting case in the second English-German example, which requires translating the *doubly-negated* phrase, "not incompatible". The attentional model correctly produces "nicht ... unvereinbar"; whereas the non-attentional model generates "nicht vereinbar", meaning "not compatible".[18] The attentional model also demonstrates its superiority in translating long sentences as in the last example.

6 Conclusion

In this paper, we propose two simple and effective attentional mechanisms for neural machine

[14] There is a subtle difference in how we retrieve alignments for the different alignment functions. At time step t in which we receive y_{t-1} as input and then compute $\boldsymbol{h}_t, \boldsymbol{a}_t, \boldsymbol{c}_t$, and $\tilde{\boldsymbol{h}}_t$ before predicting y_t, the alignment vector $\boldsymbol{a}_t$ is used as alignment weights for (a) the predicted word y_t in the *location-based* alignment functions and (b) the input word y_{t-1} in the *content-based* functions.

[15] With *concat*, the perplexities achieved by different models are 6.7 (global), 7.1 (local-m), and 7.1 (local-p).

[16] We concatenate the 508 sentence pairs with 1M sentence pairs from WMT and run the Berkeley aligner.

[17] `http://arxiv.org/abs/1508.04025`

[18] The reference uses a more fancy translation of "incompatible", which is "im Widerspruch zu etwas stehen". Both models, however, failed to translate "passenger experience".

English-German translations	
src	Orlando Bloom and Miranda Kerr still love each other
ref	Orlando Bloom und *Miranda Kerr* lieben sich noch immer
best	Orlando Bloom und *Miranda Kerr* lieben einander noch immer .
base	Orlando Bloom und **Lucas Miranda** lieben einander noch immer .
src	″ We ′ re pleased the FAA recognizes that an enjoyable passenger experience is not incompatible with safety and security , ″ said Roger Dow , CEO of the U.S. Travel Association .
ref	“ Wir freuen uns , dass die FAA erkennt , dass ein angenehmes Passagiererlebnis nicht im Widerspruch zur Sicherheit steht ” , sagte *Roger Dow* , CEO der U.S. Travel Association .
best	″ Wir freuen uns , dass die FAA anerkennt , dass ein angenehmes ist nicht mit Sicherheit und Sicherheit *unvereinbar* ist ″ , sagte *Roger Dow* , CEO der US - die .
base	″ Wir freuen uns über die <unk> , dass ein <unk> <unk> mit Sicherheit nicht **vereinbar** ist mit Sicherheit und Sicherheit ″ , sagte *Roger* **Cameron** , CEO der US - <unk> .
German-English translations	
src	In einem Interview sagte Bloom jedoch , dass er und Kerr sich noch immer lieben .
ref	However , in an interview , Bloom has said that he and *Kerr* still love each other .
best	In an interview , however , Bloom said that he and *Kerr* still love .
base	However , in an interview , Bloom said that he and **Tina** were still <unk> .
src	Wegen der von Berlin und der Europäischen Zentralbank verhängten strengen Sparpolitik in Verbindung mit der Zwangsjacke , in die die jeweilige nationale Wirtschaft durch das Festhalten an der gemeinsamen Währung genötigt wird , sind viele Menschen der Ansicht , das Projekt Europa sei zu weit gegangen
ref	The *austerity imposed by Berlin and the European Central Bank , coupled with the straitjacket* imposed on national economies through adherence to the common currency , has led many people to think Project Europe has gone too far .
best	Because of the strict *austerity measures imposed by Berlin and the European Central Bank in connection with the straitjacket* in which the respective national economy is forced to adhere to the common currency , many people believe that the European project has gone too far .
base	Because of the pressure **imposed by the European Central Bank and the Federal Central Bank with the strict austerity** imposed on the national economy in the face of the single currency , many people believe that the European project has gone too far .

Table 5: **Sample translations** – for each example, we show the source (*src*), the human translation (*ref*), the translation from our best model (*best*), and the translation of a non-attentional model (*base*). We italicize some *correct* translation segments and highlight a few **wrong** ones in bold.

translation: the *global* approach which always looks at all source positions and the *local* one that only attends to a subset of source positions at a time. We test the effectiveness of our models in the WMT translation tasks between English and German in both directions. Our local attention yields large gains of up to 5.0 BLEU over non-attentional models that already incorporate known techniques such as dropout. For the English to German translation direction, our ensemble model has established new state-of-the-art results for both WMT'14 and WMT'15.

We have compared various alignment functions and shed light on which functions are best for which attentional models. Our analysis shows that attention-based NMT models are superior to non-attentional ones in many cases, for example in translating names and handling long sentences.

Acknowledgment

We gratefully acknowledge support from a gift from Bloomberg L.P. and the support of NVIDIA Corporation with the donation of Tesla K40 GPUs. We thank Andrew Ng and his group as well as the Stanford Research Computing for letting us use their computing resources. We thank Russell Stewart for helpful discussions on the models. Lastly, we thank Quoc Le, Ilya Sutskever, Oriol Vinyals, Richard Socher, Michael Kayser, Jiwei Li, Panupong Pasupat, Kelvin Gu, members of the Stanford NLP Group and the annonymous reviewers for their valuable comments and feedback.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Christian Buck, Kenneth Heafield, and Bas van Ooyen. 2014. N-gram counts and language models from the common crawl. In *LREC*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.

Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent NN: first results. *CoRR*, abs/1412.1602.

Alexander Fraser and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. In *ICML*.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *ACL*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *NAACL*.

Minh-Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *ACL*.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent models of visual attention. In *NIPS*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2015. Recurrent neural network regularization. In *ICLR*.