Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation

Jie Zhou  Ying Cao  Xuguang Wang  Peng Li  Wei Xu
Baidu Research - Institute of Deep Learning
Baidu Inc., Beijing, China
{zhoujie01, caoying03, wangxuguang, lipeng17, wei.xu}@baidu.com

Abstract

Neural machine translation (NMT) aims at solving machine translation (MT) problems with purely neural networks and exhibits promising results in recent years. However, most of the existing NMT models are of shallow topology and there is still a performance gap between the single NMT model and the best conventional MT system. In this work, we introduce a new type of linear connections, named fast-forward connections, based on deep Long Short-Term Memory (LSTM) network, together with the interleaved bidirectional way for stacking them. Fast-forward connections play an essential role to propagate the gradients in building the deep topology of depth 16. On WMT’14 English-to-French task, we achieved BLEU=37.7 with single attention model, which outperforms the corresponding single shallow model by 6.2 BLEU points. It is the first time that a single NMT model achieves state-of-the-art performance and outperforms the best conventional model by 0.7 BLEU points. Even without considering attention mechanism, we can still achieve BLEU=36.3. After the special handling for unknown words and the model ensembling, we obtained the best score on this task with BLEU=40.4. Our models are also verified on the more difficult WMT’14 English-to-German task.

1 Introduction

Neural machine translation (NMT) has attracted a lot of interests in solving the machine translation (MT) problems in recent years (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2014). Unlike conventional statistical machine translation (SMT) systems (Koehn et al., 2003; Durrani et al., 2014) which consist of multiple separately tuned components, NMT models encode the source sequence into continuous representation space and generate the target sequence in an end-to-end form. Moreover, NMT models can also be easily adapted to other tasks such as dialog systems (Vinyals and Le, 2015), question answering systems (Yu et al., 2015) and image caption generation (Mao et al., 2014).

Generally, there are two types of NMT topologies named encoder-decoder network (Sutskever et al., 2014) and attention network (Bahdanau et al., 2014). Encoder-decoder network represents the source sequence with a fixed dimensional vector and the target sequence is generated from this vector word by word. Attention network considers the representations from all time steps of the input sequence, building a detailed relationship between the target words and input words. Recent results show that the systems based on these models can achieve similar performance with conventional SMT systems (Luong et al., 2015; Jean et al., 2015).

However, single neural models of either of the above types have not been competitive with the best conventional system (Durrani et al., 2014) when evaluated on WMT’14 English-to-French task. The best BLEU score from single model which has six layers is only 31.5 (Luong et al., 2015) while the conventional method of (Durrani et al., 2014) gives 37.0.

We focus on improving the single model per-
performance by increasing the model depth. Deep topology has been proven to outperform the shallow architecture in computer vision. In the past two years the top positions of the ImageNet contest are always occupied by the systems with more than tens or even hundreds of layers (Szegedy et al., 2015; He et al., 2015). But in NMT, the largest depth used successfully is only six (Luong et al., 2015). We attribute this problem to the properties of Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) which is widely used in NMT. In LSTM, there are more non-linear activations than those in convolution layer. These activations largely decrease the gradient in deep topology, especially when the information propagates in recurrent form. There are also many efforts to increase the depth of LSTM such as (Kalchbrenner et al., 2015), where the shortcuts do not avoid the nonlinear and recurrent computation.

In this work, we introduce the new linear connections for multi-layer recurrent networks. These connections which are called fast-forward connections play an essential role in building the deep topology with depth of 16. Besides, we also introduce the interleaved bi-directional way to stack LSTM layers in encoder part. The topology works for both encoder-decoder network and attention network. On WMT’14 English-to-French task, this is the deepest NMT topology that has ever been investigated. With our deep attention model, the BLEU score can be improved to 37.7 and outperforms the shallow model which has six layers (Luong et al., 2015) by 6.2 BLEU points. This is also the first time on this task that a single NMT model achieves state-of-the-art performance and outperforms the best conventional SMT system (Durrani et al., 2014) with the improvement of 0.7. Even without considering the attention mechanism, we can still achieve 36.3 with a single model. After model ensembling and unknown word processing, the BLEU score can be further improved to 40.4. When evaluated on the subset of test corpus without unknown words, our model gives 41.4. As a reference, previous work showed that oracle re-scoring the SMT generated 1000-best sequences exhibits the BLEU score of about 45 (Sutskever et al., 2014). Our models are also verified on the more difficult WMT’14 English-to-German task.

2 Neural Machine Translation

Neural machine translation aims at generating the target word sequence \( y = \{y_1, \ldots, y_n\} \) given the source word sequence \( x = \{x_1, \ldots, x_m\} \) with neural models. In this task, the likelihood \( p(y \mid x; \theta) \) of the target sequence will be maximized (Forcada and Neco, 1997) with parameter \( \theta \) to learn:

\[
p(y \mid x; \theta) = \prod_{j=1}^{m+1} p(y_j \mid y_{0:j-1}, x; \theta) \tag{1}
\]

\( y_{0:j-1} \) is the sub sequence from \( y_0 \) to \( y_{j-1} \). \( y_0 \) and \( y_{m+1} \) denote the start mark and end mark of target sequence respectively.

The process can be explicitly split into encoding part, decoding part and interface between these two parts. In the encoding part, the source sequence is processed and transformed into a group of vectors \( e = \{e_1, \cdots, e_m\} \) for each time step. Further operations will be used at the interface part to extract the final representation \( e \) of the source sequence from \( e \). At decoding step, the target sequence will be generated from the representation \( e \).

Recently, there are two types of NMT models which are distinct in the interface part. In the encoder-decoder model (Sutskever et al., 2014), a single vector extracted from \( e \) is used as the representation. In the attention model (Bahdanau et al., 2014), \( e \) is dynamically obtained according to the relationship between target sequence and source sequence.

Recurrent neural network (RNN) or its specific form LSTM is generally employed as the basic unit of the encoding and decoding function. However, most of the existing works are of shallow topology. In attention network, the encoding part and decoding part have only one LSTM layer respectively. In encoding-decoding network, people used at most six LSTM layers (Luong et al., 2015). As machine translation is considered to be a difficult problem, we believe more complex encoding and decoding functions are needed for modeling the relationship between the source sequence and target sequence. In this work, we focus on enhancing the complexity of the encoding/decoding functions by increasing the model depth.

Deep neural models have been studied in a wild range of problems. In computer vision, models
with more than tens of convolution layers outperform the shallow ones on a series of image tasks in recent years (Srivastava et al., 2015; He et al., 2015; Szegedy et al., 2015). Different kinds of shortcut connections are proposed to decrease the length of the gradient propagation path. Training networks based on LSTM layers which is widely used in language problems is a much more challenging task. Because of the existence of large amount of nonlinear activations and the recurrent computation, gradient values are not stable and generally are smaller. Following the same spirit for convolutional network, a lot of efforts have also been spent into training deep LSTM topologies. Yao et al. (2015) introduced depth-gated shortcuts connecting LSTM cells at adjacent layers to provide a fast way to propagate the gradients. They verified the modification of these shortcuts on MT task and language modeling task, but the best score are from models with three layers. Similarly, Kalchbrenner et al. (2015) extended this topology to be multi-dimensional. They decreased the number of nonlinear activations and path length, but the gradient propagation still relies on the recurrent computation. The investigations were also made on question-answering to encode the questions, where at most two LSTM layers are stacked (Hermann et al., 2015).

Based on the above considerations, we propose new connections to facilitate gradient propagation in the following section.

3 Deep Topology

We build the deep LSTM network with the new proposed linear connections. The shortest paths through the proposed connections do not include any nonlinear transformations and do not rely on any recurrent computation. We call these connections fast-forward connections. Within the deep topology, we also introduce the interleaved bi-directional way to stack the LSTM layers. And the way to extract the source sequence representation is consequently modified.

3.1 Network

Our whole deep neural network is shown in Fig. 2. This topology can be divided into three parts: encoder part (P-E) on the left, decoder part (P-D) on the right and the interface between these two parts (P-I) which extracts the representation of the source sequence. We have two instantiations of this topology named Deep-ED and Deep-Att, corresponding to the extension of encoder-decoder network and attention network respectively. Our essential innovation is made for the adjacent stacked recurrent layers and we will start with the basic RNN model for the sake of clarity.

Recurrent layer: When an input sequence \( \{x_1, \ldots, x_m\} \) is given to a recurrent layer, the output \( h_t \) at each time step \( t \) can be computed as (see Fig. 1 (a))

\[
h_t = \sigma(W_f x_t + W_r h_{t-1}) = \text{RNN} (f_t, h_{t-1}), \tag{2}
\]

where bias parameter is not included for simplicity. We use red circle, blue empty square to denote the input and hidden state. Blue square with “-” denotes previous hidden state. Dotted line means that the hidden state is recurrently used. This computation can be equivalently split into two consecutive steps:

- **Feed-Forward computation:** \( f_t = W_f x_t \). Left part in Fig. 1 (b). “\( f \)” block.
- **Recurrent computation:** \( \text{RNN} (f_t, h_{t-1}) \). Right part and the sum operation (+) followed by activation in Fig. 1 (b). “\( r \)” block.

For a deep topology with stacked recurrent layers, the input of each block \( f \) at recurrent layer \( k \) (denoted by \( f^k \)) is usually the output of block \( r \) at its previous recurrent layer \( k - 1 \) (denoted by \( h^{k-1} \)). In our work, we add fast-forward connections (F-F connections) which connect two feed-forward computation blocks \( f \) of adjacent recurrent layers. It means that each block \( f \) at recurrent layer \( k \) takes both outputs of block \( f \) and block \( r \) at its previous layer as input (Fig. 1 (c)). F-F connections are denoted by dashed red lines in Fig. 1 (c) and Fig. 2. The path of F-F connections contains neither nonlinear activations nor recurrent computation. It provides a fast way for information to travel, so we call this path fast-forward connections.

Additionally, in order to learn more temporal dependencies, the sequences could be processed in
different directions at each pair of adjacent recurrent layers. This is quantitatively expressed in Eq. 3:

\[
\begin{align*}
 f^k_t &= W^k_f \cdot [f^{k-1}_t, h^{k-1}_t], \quad k > 1 \\
 f^1_t &= W^1_f x_t \\
 h^k_t &= \text{RNN}^k \left( f^k_t, h^{k-1}_{t+(-1)^k} \right)
\end{align*}
\] (3)

The opposite directions are marked by direction term \((-1)^k\). At the beginning layer, the block f takes \(x_t\) as the input. \([\cdot, \cdot]\) denotes the concatenation of the inside vectors. This is shown in Fig. 1 (c). Two variations are summarized here:

- We add a connection between \(f^k_t\) and \(f^{k-1}_t\). Without \(f^{k-1}_t\), our model will be reduced to the traditional stacked model.
- We alternate the RNN direction at different layer \(k\) with the direction term \((-1)^k\). If we fix the direction term to \(-1\), all layers work in the forward direction.

**LSTM layer:** Actually, a specific type recurrent layer called LSTM (Hochreiter and Schmidhuber, 1997; Graves et al., 2009) is used in our work. LSTM is structurally more complex than basic RNN used in Eq. 2. We define the computation of LSTM as a function which maps the input \(f\) and its state-output pair \((h, s)\) at previous time step to a new current state-output pair. The exact computations for \((h_t, s_t) = \text{LSTM}(f_t, h_{t-1}, s_{t-1})\) are the following:

\[
\begin{align*}
 [z, z_\rho, z_\phi, z_\pi] &= f_t + W_r h_{t-1} \\
 s_t &= \sigma_i(z) \circ \sigma_g(z + s_{t-1} \circ \theta_\rho) + \\
 &\quad \sigma_g(z_\rho + s_{t-1} \circ \theta_\phi) \circ s_{t-1} \\
 h_t &= \sigma_o(s_t) \circ \sigma_g(z_\pi + s_t \circ \theta_\pi)
\end{align*}
\] (4)

where \([z, z_\rho, z_\phi, z_\pi]\) is the concatenation of four vectors of equal size, \(\circ\) means element-wise multiplication, \(\sigma_i\) is the input activation function, \(\sigma_o\) is the output activation function, \(\sigma_g\) is the activation function for gates, and \(W_r, \theta_\rho, \theta_\phi, \text{ and } \theta_\pi\) are the parameters of the LSTM. It is slightly different from the standard notation in that we do not have a matrix to multiply with the input \(f\) in our notation.

With this notation, we can write down the computations for our deep bi-directional LSTM model with F-F connections:

\[
\begin{align*}
 f^k_t &= W^k_f \cdot [f^{k-1}_t, h^{k-1}_t], \quad k > 1 \\
 f^1_t &= W^1_f x_t, \\
 (h^k_t, s^k_t) &= \text{LSTM}^k \left( f^k_t, h^k_{t+(-1)^k}, s^k_{t+(-1)^k} \right)
\end{align*}
\] (5)

where \(x_t\) is the input to the deep bi-directional LSTM model. For encoder, \(x_t\) is the embedding of \(t^{th}\) word in the source sentence. For decoder \(x_t\) is the concatenation of the embedding of the \(t^{th}\) word in the target sentence and the encoder representation for step \(t\).

In our final model two additional operations are used with Eq. 5, which is shown in Eq. 6. Half(f) means the first half elements of \(h\), and Dr(h) is the dropout operation (Hinton et al., 2012) which randomly set an element of \(h\) to zero with a certain probability. The use of Half(\cdot) is to reduce the parameter size and does not affect on the performance. While we tried to only use the first thirds parameters, the performance degrades considerably.

\[
 f^k_t = W^k_f \cdot [\text{Half}(f^{k-1}_t), \text{Dr}(h^{k-1}_t)], \quad k > 1
\] (6)

With the F-F connections, we build a channel to propagate the gradients in deep topology. F-F connections will accelerate the model convergence and meanwhile improve the performance. Similar idea
Figure 2: Network. It includes three parts from left to right: encoder part (P-E), interface (P-I) and decoder part (P-D). We only show the topology of Deep-Att as an example. “f” and “r” blocks correspond to feed-forward part and the following LSTM computation. The F-F connections are denoted by red lines.

was also used in (He et al., 2015; Zhou and Xu, 2015).

**Encoder:** The LSTM layers are stacked following Eq. 5. We call this type of encoder **interleaved bi-directional encoder**. Besides, there are two similar columns ($a_1$ and $a_2$) in encoder part. Each column consists of $n_e$ stacked LSTM layers. There is no connection between the columns. The first layers of both columns process the word representations of the source sequence in different direction. At the last LSTM layers, there are two groups of vectors representing the source sequence. The group size is the same as the length of the input sequence.

**Interface:** Previous encoder-decoder model and attention model are different in how to extract the representations of the source sequences. In our work, as a consequence of the introduced F-F connections, we have 4 output vectors ($h^k_t$ and $f^k_t$ of both columns). The representations are modified for both Deep-ED and Deep-Att.

For Deep-ED, $e_t$ is static and consists of four parts. 1: The last time step output $h^k_m$ of the first column. 2: Max-operation $\text{Max}(\cdot)$ of $h^k_t$ at all time steps of the second column, denoted by $\text{Max}(h^{n_e,a_2}).$ $\text{Max}(\cdot)$ means to obtain max value from each dimension over $t$. 3: $\text{Max}(f^{n_e,a_1}).$ 4: $\text{Max}(f^{n_e,a_2}).$ Max-operation and last time step extraction provide complimentary information but does not affect the performance much. $e_t$ is used as the final representation $c_t$.

For Deep-Att, we do not need the above two operations. We only concatenate the 4 output vectors at each time steps to obtain $e_t$, and a soft attention mechanism (Bahdanau et al., 2014) is used to calculate the final representation $c_t$ from $e_t$. $e_t$ is summarized as:

Deep-ED: $e_t$

$$[h^{n_e,a_2}_m, \text{Max}(h^{n_e,a_2}_t), \text{Max}(f^{n_e,a_1}_t), \text{Max}(f^{n_e,a_2}_t)]$$

Deep-Att: $e_t$

$$[h^{n_e,a_1}_t, h^{n_e,a_2}_t, f^{n_e,a_1}_t, f^{n_e,a_2}_t]$$  \hspace{1cm} (7)

Note that the vector dimension of $f$ is four times larger than that of $h$ (see Eq. 4). $c_t$ is summarized as:

Deep-ED: $c_t = e_t$, \hspace{0.5cm} (const)

Deep-Att: $c_t = \sum_{t'=1}^{m} \alpha_{t,t'} W_p e_{t'}$  \hspace{1cm} (8)

$\alpha_{t,t'}$ is the normalized attention weight computed by:

$$\alpha_{t,t'} = \frac{\exp(a(W_p e_{t'}, h^{1,dec}_{t-1}))}{\sum_{t''} \exp(a(W_p e_{t''}, h^{1,dec}_{t-1}))}$$  \hspace{1cm} (9)

$h^{1,dec}_{t-1}$ is the first hidden layer output in decoding part. $a(\cdot)$ is an alignment model described in (Bahdanau et al., 2014). For Deep-Att to save the memory cost, we linearly project (with $W_p$) the concatenated vector $e_t$ to a vector with 1/4 dimension size, denoted by fc (fully connected) block in Fig. 2.
**Decoder:** The decoder follows the Eq. 5 and Eq. 6 with fixed direction term $-1$. At the first layer, we use the following $x_t$:

$$x_t = [c_t, y_{t-1}]$$

(10)

$y_{t-1}$ is the target word embedding at the previous time step and initially $y_0$ is zero. There is a single column of $n_d$ stacked LSTM units. We also use the F-F connections as those in encoder and all layers work in forward direction. Note that at the last LSTM layer, we only use $h_t$ to make the prediction with a softmax layer.

### 3.2 Training technique

We take the parallel data as the only input without using any monolingual data for neither word representation pre-training nor language modeling. Because of the deep bi-directional structure, we do not need to reverse the sequence order as (Sutskever et al., 2014).

The deep topology brings difficulties into the model training, especially when the first order methods such as stochastic gradient descent (SGD) (LeCun et al., 1998) is used. The parameters should be properly initialized and the converging process would be slow. We tried several high order optimization techniques such as AdaDelta (Zeiler, 2012), RMSProp (Tieleman and Hinton, 2012) and Adam (Kingma and Ba, 2014). We found all of them are able to speed up the process a lot compared to simple SGD while no significant difference is observed among them in performance. In this work, we choose Adam for model training and will not compare their detailed behavior.

Dropout (Hinton et al., 2012) is also used to avoid over-fitting. It is utilized on LSTM nodes $h_t^L$ (See Eq. 5) with ratio of $p_d$ for both encoder and decoder.

During the whole model training process, we keep all hyper parameters fixed without any intermediate interruption. The hyper parameters are selected according to the performance on development set. For such a deep and large network, it is not easy to determine the tuning strategy and this would be considered in future work.

### 3.3 Generation

We use the common left-to-right beam-search method for sequence generation. At each time step $t$, the word $y_t$ can be predicted by:

$$\hat{y}_t = \arg \max_y P(y_t|\hat{y}_{0:t-1}, x; \theta)$$

(11)

where $\hat{y}_t$ is the predicted target word, $\hat{y}_{0:t-1}$ is the generated sequence from time step 0 to $t - 1$. We keep $n_b$ best candidates according to Eq. 11 at each time step, until the end mark is generated. The hypothesis are ranked by the total likelihood of the generated sequence, despite in some works normalized likelihood is considered (Jean et al., 2015).

### 4 Experiment

We evaluate our method mainly on the widely used WMT’14 English-to-French translation task. In order to verify our model on more difficult language pairs, we also give the results of WMT’14 English-to-German translation task.

#### 4.1 Data sets

For both tasks, we use the full WMT’14 parallel corpus as our training data. The detailed data sets are listed below:

- **English-to-French:** Europarl v7, Common Crawl, UN, News Commentary, Gigaword
- **English-to-German:** Europarl v7, Common Crawl, News Commentary

In total, the English-to-French corpus includes 36 million sentence pairs, and English-to-German corpus includes 4.5 million sentence pairs. The news-test-2012 and news-test-2013 are concatenated as our development set, and the news-test-2014 is the test set. Our data set is consistent with the previous works on NMT (Luong et al., 2015; Jean et al., 2015) to ensure fair comparison.

For the source language, we select the most frequent 200K words as the input vocabulary. For the target language we select the most frequent 80K French words and 160K German words as the output vocabulary. The full vocabulary of German corpus is larger (Jean et al., 2015), so we select more German words to build the target vocabulary. Out-of-vocabulary words are replaced with the unknown symbol $\langle\text{unk}\rangle$. For complete comparison with previous work on English-to-French task, we also show the results with small vocabulary of 30K input words.
and 30K output words on the sub train set with selected 12M parallel sequences (Schwenk, 2014; Sutskever et al., 2014; Cho et al., 2014).

4.2 Model settings
We have two models as described above, named Deep-ED and Deep-Att. Both models have exactly the same configuration and layer size except the interface part P-I.

We use 256 dimensional word embeddings for both source and target languages. All LSTM layers, including $2 \times n_c$ layers in encoder and $n_d$ layers in decoder, have 512 memory cells. The output layer size is the same as the size of the target vocabulary. The dimension of $c_t$ is 5120 and 1280 for Deep-ED and Deep-Att respectively. For each LSTM layer, activation functions for gates, inputs and outputs are sigmoid, tanh, and tanh respectively.

Our network is narrow on word embeddings and LSTM layers. Note that in previous work (Sutskever et al., 2014; Bahdanau et al., 2014), 1000 dimensional word embeddings and 1000 dimensional LSTM layers are used. We also tried larger scale models but did not obtain further improvements. Consider comparing the computation complexity with Luong et al. (2015), ours might be much lower than theirs.

4.3 Optimization
Note that each LSTM layer includes two parts as described in Eq. 3, feed-forward computation and recurrent computation. Since there are non-linear activations in the recurrent computation, a larger learning rate $l_r = 5 \times 10^{-4}$ is used while for feed-forward computation a smaller learning rate $l_f = 4 \times 10^{-5}$ is used. Word embeddings and softmax layer also use this learning rate $l_f$. We refer to all parameters not used for recurrent computation as non-recurrent part of the model.

Because of the large model size, we use strong $L_2$ regularization to constrain the parameter matrix $v$ in the following way:

$$v \leftarrow v - l \cdot (g + r \cdot v)$$ (12)

Here $r$ is the regularization strength, $l$ is the corresponding learning rate, $g$ stands for the gradients of $v$. Two embedding layers are not regularized. All the other layers have the same $r = 2$.

Parameters of the recurrent computation part are set to be zero at the beginning. All non-recurrent parts are randomly initialized with zero mean and standard deviation of 0.07. A detailed guide for setting hyper-parameters can be found in Bengio (2012).

Dropout ratio $p_d$ is set to be 0.1. In each batch, there are 500 $\sim$ 800 sequences in our work. The exact number depends on the sequence lengths and model size. We also find that larger batch size results in better convergence although the improvement is not large. But this is constrained by the GPU memory. We need 4 $\sim$ 8 GPU machines (each has 4 K40 GPU cards) running for 10 days to train the full model with parallelization at data batch level. It takes nearly 1.5 days for each pass.

One thing we want to stress here is that our deep model is not sensitive to these settings. Small variations around this region will not affect the final performance.

4.4 Results
We keep the same evaluation rule with most previous NMT works (Sutskever et al., 2014; Luong et al., 2015; Jean et al., 2015). All reported BLEU scores are computed with the multi-bleu.perl script which is also used in the above works. The results are tokenized and case sensitive.

4.4.1 Single models

**English-to-French:** First we list our single model results on English-to-French task in Tab. 1. In the first block we show the results with the full corpus. The previous best single NMT encoder-decoder model (Enc-Dec) with six layers gives BLEU=31.5 (Sutskever et al., 2014). From Deep-ED, we obtain the BLEU score of 36.3, which outperforms Enc-Dec model by 4.8 BLEU points. This result is even better than the ensemble result of eight Enc-Dec models which is 35.6 (Luong et al., 2015). This shows that deep topology can also works well with LSTM layer, besides convolutional layers in computer vision. For Deep-Att, the performance is further improved to be 37.7. We also list the previous state-of-the-art performance from conventional SMT system (Durrani et al., 2014) with BLEU

\[\text{BLEU} \approx \frac{1}{K} \sum_{k=1}^{K} \frac{P^k}{C^k} \]
of 37.0. This is the first time that a single NMT model trained in end-to-end form beats the best conventional system.

We also show the results on the smaller data set with 12M sentence pairs and 30K vocabulary in the second block. Two attention models RNNsearch (Bahdanau et al., 2014) and RNNsearch-LV (Jean et al., 2015) give the BLEU score of 28.5 and 32.7 respectively. Note that RNNsearch-LV uses a large output vocabulary of 500K words based on the standard attention model RNNsearch. We obtain BLEU=35.9 which outperforms its corresponding shallow model RNNsearch by 7.4 BLEU points. The SMT result from (Schwenk, 2014) is also listed and falls behind our model by 2.6 BLEU points.

Table 1: English-to-French task: BLEU scores of single neural models. We also list the conventional SMT system.

| Methods          | Data | Voc | BLEU |
|------------------|------|-----|------|
| Enc-Dec (Luong, 2015) | 36M  | 80K | 31.5 |
| SMT (Durrani, 2014) | 36M  | Full | 37.0 |
| Deep-ED (Ours)   | 36M  | 80K | 36.3 |
| Deep-Att (Ours)  | 36M  | 80K | 37.7 |
| RNNsearch (Bahdanau, 2014) | 12M  | 30K | 28.5 |
| RNNsearch-LV (Jean, 2015) | 12M | 500K | 32.7 |
| SMT (Schwenk, 2014) | 12M  | Full | 33.3 |
| Deep-Att (Ours)  | 12M  | 30K | 35.9 |

Besides, during the generation process, we obtained the best BLEU score with beam size = 3 (when beam size is 2, there is only 0.1 difference in BLEU score.). This is different from other works listed in Tab. 1, where the beam size is 12 (Jean et al., 2015; Sutskever et al., 2014). We attribute this difference to the improved model performance, where the ground truth generally exists in the top hypothesis. Consequently, with largely decreased beam size, the generation efficiency is significantly improved.

Next we are going to list the effect of the novel F-F connections in our Deep-Att model of shallow topology in Tab. 2. When \( n_e = 1 \) and \( n_d = 1 \), the BLEU scores are 31.2 without F-F and 32.3 with F-F. Note that the model without F-F is exactly the standard attention model (Bahdanau et al., 2014). Since here is only a single layer, F-F connections means at the interface part we include \( f_i \) into the representation (see Eq. 7). We find F-F connections bring 1.1 improvement of BLEU score. After we increase our model depth to \( n_e = 2 \) and \( n_d = 2 \), the improvement is enlarged to 1.4. When the model is trained with larger depth without F-F connections, we find that the parameter exploding problem (Bengio et al., 1994) happens too frequently that we could not finish training. This suggests that F-F connections provide a fast way for gradient propagation.

Table 2: The effect of F-F. We list the BLEU scores of Deep-Att with and without F-F. Because of the parameter exploding problem, we can not list the model performance of larger depth without F-F. For \( n_e = 1 \) and \( n_d = 1 \), F-F connections only contribute to the representation at interface part (see Eq. 7).

| Models          | F-F | \( n_e \) | \( n_d \) | BLEU |
|-----------------|-----|----------|----------|------|
| Deep-Att        | No  | 1        | 1        | 31.2 |
| Deep-Att        | Yes | 1        | 1        | 32.3 |
| Deep-Att        | No  | 2        | 2        | 33.3 |
| Deep-Att        | Yes | 2        | 2        | 34.7 |

Removing F-F connections also reduces the corresponding model size. In order to figure out the effect of F-F between models with the same parameter size, we increase the LSTM layer width of Deep-Att without F-F. In Tab. 3 we show that, after using two times larger LSTM layer width of 1024, we can only obtain BLEU score of 33.8. It is still behind the corresponding Deep-Att with F-F.

Table 3: BLEU scores with different LSTM layer width in Deep-Att. After using two times larger LSTM layer width of 1024, we can only obtain BLEU score of 33.8. It is still behind the corresponding Deep-Att with F-F.

| Models          | F-F | \( n_e \) | \( n_d \) | Width | BLEU |
|-----------------|-----|----------|----------|-------|------|
| Deep-Att        | No  | 2        | 2        | 512   | 33.3 |
| Deep-Att        | No  | 2        | 2        | 1024  | 33.8 |
| Deep-Att        | Yes | 2        | 2        | 512   | 34.7 |

We also notice that the interleaved bi-directional encoder start to work when the encoder depth is larger than 1. This property is investigated in Tab. 4. For our largest model with \( n_e = 9 \) and \( n_d = 7 \), we compared the BLEU scores of interleaved bi-directional encoder and uni-directional encoder (all LSTM layers work in forward direction). We find there is a gap about 1.5 points between these two encoders in both Deep-Att and Deep-ED.
Table 4: The effect of interleaved bi-directional encoder. We list the BLEU scores of our largest Deep-Att and Deep-ED. The encoder term Bi denotes the interleaved bi-directional encoder is used. Uni denotes all LSTM layers work in forward direction.

| Models  | Encoder | $n_e$ | $n_d$ | BLEU |
|---------|---------|-------|-------|------|
| Deep-Att | Bi      | 9     | 7     | 37.7 |
| Deep-Att | Uni     | 9     | 7     | 36.2 |
| Deep-ED  | Bi      | 9     | 7     | 36.3 |
| Deep-ED  | Uni     | 9     | 7     | 34.9 |

Table 5: BLEU score of Deep-Att with different model depth. With $n_e = 1$ and $n_d = 1$, F-F connections only contribute to the representation at interface part where $f_t$ is included (see Eq. 7).

| Models  | F-F   | $n_e$ | $n_d$ | Col | BLEU |
|---------|-------|-------|-------|-----|------|
| Deep-Att | Yes   | 1     | 1     | 2   | 32.3 |
| Deep-Att | Yes   | 2     | 2     | 2   | 34.7 |
| Deep-Att | Yes   | 5     | 3     | 2   | 36.0 |
| Deep-Att | Yes   | 9     | 7     | 2   | 37.7 |
| Deep-Att | Yes   | 9     | 7     | 1   | 36.6 |

Table 6: Comparison of encoders with different number of columns and LSTM layer width.

| Models  | F-F   | $n_e$ | $n_d$ | Col | width | BLEU |
|---------|-------|-------|-------|-----|-------|------|
| Deep-Att | Yes   | 9     | 7     | 2   | 512   | 37.7 |
| Deep-Att | Yes   | 9     | 7     | 1   | 512   | 36.6 |
| Deep-Att | Yes   | 9     | 7     | 1   | 1024  | 36.7 |

Next we are going to look into the model depth. In Tab. 5, starting from $n_e = 1$ and $n_d = 1$ and gradually increasing the model depth, we significantly increase BLEU scores. With $n_e = 9$ and $n_d = 7$, the best scores for Deep-Att is 37.7. We tried to increase the LSTM width based on this, but obtained little improvements. As we stated in Sec.2, the complexity of encoding/decoding functions, which is related to the model depth, is more important than model size. We also tried larger depth further, but the results start to get worse. With our topology and training technique, $n_e = 9$ and $n_d = 7$ is the best depth we can achieve.

Table 7: English-to-German task: BLEU scores of single neural models. We also list the conventional SMT system for comparison.

| Methods            | Data  | Voc  | BLEU |
|--------------------|-------|------|------|
| RNNsearch (Jean,2015) | 4.5M  | 50K  | 16.5 |
| RNNsearch-LV (Jean,2015) | 4.5M  | 500K | 16.9 |
| SMT (Buck,2014)     | 4.5M  | Full | 20.7 |
| Deep-Att (Ours)     | 4.5M  | 160K | 20.6 |

4.4.2 Post processing

Two post processing techniques are used to improve the performance further on English-to-French task.

First, three Deep-Att models are built for ensemble results. They are initialized with different parameters. And the training corpus for these models are shuffled with different random seed. We sum over the distribution of the predicted words and normalize the final distribution to generate the next word. It is shown in Tab. 8 that the model ensemble can improve the performance further to 38.9. In (Luong et al., 2015) and (Jean et al., 2015) there are eight models for the best scores, but we only use three models and we do not obtain further gain from

English-to-German: We also verify our deep
more models.

| Methods | Model            | Data  | Voc | BLEU |
|---------|------------------|-------|-----|------|
| Deep-ED | Single           | 36M   | 80K | 36.3 |
| Deep-Att| Single           | 36M   | 80K | 37.7 |
| Deep-Att| Single+PosUnk    | 36M   | 80K | 39.2 |
| Deep-Att| Ensemble         | 36M   | 80K | 38.9 |
| Deep-Att| Ensemble+PosUnk  | 36M   | 80K | 40.4 |
| SMT     | Durrani, 2014    | 36M   | Full| 37.0 |
| Enc-Dec | Ensemble+PosUnk  | 36M   | 80K | 37.5 |

Table 8: BLEU scores of different models. The first two blocks are our results of two single models and models with post processing. In the last block we list two baselines of the best conventional SMT system and NMT system.

Second, we recover the unknown words in the generated sequences with Positional Unknown (PosUnk) model introduced in (Luong et al., 2015). The full parallel corpus is used to obtain the word mappings (Liang et al., 2006). We find this method provides additional 1.5 BLEU points, consistent with the conclusion in (Luong et al., 2015). We obtain the new BLEU score of 39.2 with single Deep-Att. For the ensemble models of Deep-Att, the BLEU score rises to 40.4. At the last two lines, we list the conventional SMT model (Durrani et al., 2014) and previous best neural models based system Encoding-Decoding (Luong et al., 2015) for comparison. We find our best score outperforms the previous best score by nearly 3 points.

4.5 Analysis

4.5.1 Length

On English-to-French task, we analyze the effect of the source sequence length on our models as shown in Fig. 3. Here we show five curves of our Deep-Att single model, our Deep-Att ensemble model, our Deep-ED model, previous Enc-Dec model with four layers (Sutskever et al., 2014) and SMT model (Durrani et al., 2014). We find our Deep-Att model works better than the previous two models (Enc-Dec and SMT) on nearly all length scales. It is also shown that for very long sequences with length over 70 words, the performance of our Deep-Att does not degrade, when compared with another NMT model Enc-Dec. Our Deep-ED also have much better performance than the shallow Enc-Dec model on nearly all length scales, although for long sequences it degrades and starts to fall behind Deep-Att.

4.5.2 Unknown words

Next we look into the detail of the effect of unknown words on English-to-French task. We select the subset without unknown words on target sentence from the original test set. There are 1705 sequences (56.8%). We compute the BLEU scores on this subset and the results are shown in Tab. 9. We also list the results from SMT model (Durrani et al., 2014) as comparison.

We find the BLEU score of Deep-Att on subset rises to 40.3. Considering the score on full test set is 37.7, we have a gap of 2.4. On this subset, the SMT model gives 37.5, and it should be noted that its score on full set is 37.0. This suggests that the difficulty on this subset is not much different from the full set. So we attribute the larger gap for Deep-Att to the existence of unknown words. We also compute the BLEU score on the subset with ensemble model and obtain 41.4. As a reference related to human level, in (Sutskever et al., 2014), it has been tested that the BLEU score of oracle re-scoring the LIUM 1000-best results (Schwenk, 2014) is 45.

4.5.3 Over-fitting

Deep models have more parameters, and thus have stronger ability to fit the large data set. Besides, we will also show that deep models are less probable to over-fit the data set.
However, the committee recommended the FAA allow pilots to order passengers to shut off devices during instrument landings in low visibility.

Toutefois, le comité a recommandé que la FAA autorise les pilotes à demander aux passagers d’éteindre leurs appareils pendant les atterrissages aux instruments dans des conditions de faible visibilité.

Toutefois, le comité a recommandé que la FAA autorise les pilotes à ordonner aux passagers d’arrêter les dispositifs pendant les atterrissages d’instruments en basse visibilité.

Toutefois, le comité a recommandé la FAA permettent aux pilotes de commander les passagers à éteindre leurs appareils durant les débarquements de l’instrument de faible visibilité.

New airliners are far more reliant on electrical systems than previous generations of aircraft, but they are also designed and approved by the FAA to be resistant to electronic interference.

Les nouveaux avions de ligne dépendent beaucoup plus des systèmes électriques que les générations précédentes d’avions, mais ils sont également conçus pour résister aux interférences électroniques et approuvés par la FAA.

Les nouveaux avions de ligne dépendent beaucoup plus des systèmes électriques que les générations précédentes, mais ils sont aussi conçus et approuvés par la FAA pour résister aux interférences électroniques.

Les nouveaux avions de ligne sont bien plus dépendantes des systèmes électriques que les générations précédentes d’avion, mais ils sont aussi conçus et approuvés par la FAA pour résister aux interférences électroniques.

| Source | However, the committee recommended the FAA allow pilots to order passengers to shut off devices during instrument landings in low visibility. |
| Ref | Toutefois, le comité a recommandé que la FAA autorise les pilotes à demander aux passagers d’éteindre leurs appareils pendant les atterrissages aux instruments dans des conditions de faible visibilité. |
| Ours | Toutefois, le comité a recommandé que la FAA autorise les pilotes à ordonner aux passagers d’arrêter les dispositifs pendant les atterrissages d’instruments en basse visibilité. |
| Durrani | Toutefois, le comité a recommandé la FAA permettent aux pilotes de commander les passagers à éteindre leurs appareils durant les débarquements de l’instrument de faible visibilité. |

New airliners are far more reliant on electrical systems than previous generations of aircraft, but they are also designed and approved by the FAA to be resistant to electronic interference.

Les nouveaux avions de ligne dépendent beaucoup plus des systèmes électriques que les générations précédentes d’avions, mais ils sont également conçus pour résister aux interférences électroniques et approuvés par la FAA.

Les nouveaux avions de ligne dépendent beaucoup plus des systèmes électriques que les générations précédentes, mais ils sont aussi conçus et approuvés par la FAA pour résister aux interférences électroniques.

Les nouveaux avions de ligne sont bien plus dépendantes des systèmes électriques que les générations précédentes d’avion, mais ils sont aussi conçus et approuvés par la FAA pour résister aux interférences électroniques.

Table 10: Two Examples of English-to-French translation. For each example, there are four lines corresponding to source language, reference(ref), our ensemble model result with PosUnk and conventional SMT model (Durrani).

| Model       | Test set | Ratio(%) | BLEU |
|-------------|----------|----------|------|
| Deep-Att    | Full     | 100.0    | 37.7 |
| Ensemble    | Full     | 100.0    | 38.9 |
| SMT(Durrani)| Full     | 100.0    | 37.0 |
| Deep-Att    | Subset   | 56.8     | 40.3 |
| Ensemble    | Subset   | 56.8     | 41.4 |
| SMT(Durrani)| Subset   | 56.8     | 37.5 |

Table 9: BLEU scores of the subset of the test set without considering unknown words.

| Model       | Test set | Ratio(%) | BLEU |
|-------------|----------|----------|------|
| Deep-Att    | Full     | 100.0    | 37.7 |
| Ensemble    | Full     | 100.0    | 38.9 |
| SMT(Durrani)| Full     | 100.0    | 37.0 |
| Deep-Att    | Subset   | 56.8     | 40.3 |
| Ensemble    | Subset   | 56.8     | 41.4 |
| SMT(Durrani)| Subset   | 56.8     | 37.5 |

In Fig. 4, on English-to-French task, we show three results from models with different depth. These three models are evaluated by token error rate, which is defined as the ratio of incorrectly predicted words in the whole target sequence with correct historical input. The curve with square marks corresponds to Deep-Att with $n_e = 9$ and $n_d = 7$. The curve with circle marks corresponds to $n_e = 5$ and $n_d = 3$. The curve with triangle marks corresponds to $n_e = 1$ and $n_d = 1$. We find deep model has better performance on test set when the token error rate is the same as that of shallow models on train set. It shows that, with decreased token error rate, deep model is more advantageous in avoiding the over-fitting phenomenon. We only plot the early training stage curves because on late training stage the curves become not smooth.

4.5.4 Examples

On English-to-French task, we select two example sequences showing in Tab. 10. For each example, there are four lines corresponding to source language, reference(ref), our ensemble model result with PosUnk and conventional SMT model (Durrani) (Durrani et al., 2014) respectively.
5 Conclusion

With the introduction of fast-forward connections to the deep LSTM network, we build a fast path with neither non-linear transformations nor recurrent computation to propagate the gradients from the top to the deep bottom. On this path, gradients decays much slower compared to the standard LSTM network. This enable us to build the deep topology of NMT models.

We trained NMT models with depth of 16 including 25 LSTM layers and evaluated them mainly on WMT’14 English-to-French translation task. This is the deepest topology that have been investigated in NMT area on this task. We showed that our Deep-Att exhibits 6.2 BLEU points improvement over the previous best single model, achieving 37.7 BLEU score. This single end-to-end NMT model outperforms the conventional best SMT system (Durrani et al., 2014) and achieves the state-of-the-art performance. After utilizing the unknown word processing and model ensemble of three models, we obtained the BLEU score of 40.4, improved by 2.9 BLEU score over previous best result. When evaluated on the subset of test corpus without unknown words, our model gives 41.4. As a reference number, the work (Sutskever et al., 2014) show that oracle re-scoring the SMT generated 1000-best lists exhibits the BLEU score of about 45. Our model is also verified on the more difficult English-to-German task.

Besides, our model is efficient in sequence generation. The best results from both single model and model ensemble are obtained with beam size of 3, much smaller than previous NMT systems where beam size is about 12 (Jean et al., 2015; Sutskever et al., 2014). From our analysis, we find deep models are more advantageous at learning long sequences and the deep topology is resistant to the over-fitting phenomenon.

We tried deeper models and did not obtain further improvements with current topology and training techniques. However, 16 is not a large number of model depth when compared to the models in computer vision (He et al., 2015). We believe we can benefit from deeper models with new design of topologies and training techniques which are left as our future work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166.

Yoshua Bengio. 2012. Practical recommendations for gradient-based training of deep architectures. arXiv:1206.5533.

Christian Buck, Kenneth Heafield, and Bas van Ooyen. 2014. N-gram counts and language models from the common crawl. In Proceedings of the Language Resources and Evaluation Conference.

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 Proceedings of the Empirical Methods in Natural Language Processing (EMNLP2014), pages 48–54. October.

Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. 2014. Edinbughs phrase-based machine translation systems for WMT-14. In Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 97–104, June.

Mikel L. Forcada and Ramón P. Ñeco, 1997. Biological and Artificial Computation: From Neuroscience to Technology: International Work-Conference on Artificial and Natural Neural Networks, IWANN’97 Lanzarote, Canary Islands, Spain, June 4–6, 1997 Proceedings, chapter Recursive hetero-associative memories for translation, pages 453–462. Springer Berlin Heidelberg, Berlin, Heidelberg.

Alex Graves, Marcus Liwicki, Santiago Fernandez, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. 2009. A novel connectionist system for unconstrained handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5):855–868.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. arXiv:1512.03385.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. arXiv:1506.03340.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580.
Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, July.

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, October.

Nal Kalchbrenner, Ivo Danihelka, and Alex Graves. 2015. Grid long short-term memory. *arXiv:1507.01526*.

Diederik P. Kingma and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Volume 1, NAACL ’03*, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL ’06*, pages 104–111, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, July.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan L. Yuille. 2014. Deep captioning with multimodal recurrent neural networks (m-RNN). *arXiv:1412.6632*.

Holger Schwenk. 2014. http://www-lium.univ-lemans.fr/~schwenk/cslm_joint_paper [online; accessed 03-september-2014]. *University Le Mans*.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv:1505.00387*.

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

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv:1506.05869*.

Kaisheng Yao, Trevor Cohn, Katerina Vylomova, Kevin Duh, and Chris Dyer. 2015. Depth-gated LSTM. *arXiv:1508.03790*.

Yang Yu, Wei Zhang, Chung-Wei Hang, Bing Xiang, and Bowen Zhou. 2015. Empirical study on deep learning models for QA. *arXiv:1510.07526*.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. *arXiv:1212.5701*.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137, July.