Neural Machine Translation with Reconstruction

Zhaopeng Tu† Yang Liu† Lifeng Shang† Xiaohua Liu† Hang Li†

†Noah’s Ark Lab, Huawei Technologies, Hong Kong
{tu.zhaopeng,shang.lifeng,liuxiaohua3,hangli.hl}@huawei.com
‡Department of Computer Science and Technology, Tsinghua University, Beijing
liuyang2011@tsinghua.edu.cn

Abstract

Although end-to-end Neural Machine Translation (NMT) has achieved remarkable progress in the past two years, it suffers from a major drawback: translations generated by NMT systems often lack of adequacy. It has been widely observed that NMT tends to repeatedly translate some source words while mistakenly ignoring other words. To alleviate this problem, we propose a novel encoder-decoder-reconstructor framework for NMT. The reconstructor, incorporated into the NMT model, manages to reconstruct the input source sentence from the hidden layer of the output target sentence, to ensure that the information in the source side is transformed to the target side as much as possible. Experiments show that the proposed framework significantly improves the adequacy of NMT output and achieves superior translation result over state-of-the-art NMT and statistical MT systems.

Introduction

Past several years have observed a significant progress in Neural Machine Translation (NMT) (Kalchbrenner and Blunsom 2013; Cho et al. 2014; Sutskever et al. 2014; Bahdanau et al. 2015). Particularly, NMT has significantly enhanced the performance of translation between a language pair involving rich morphology prediction and/or significant word reordering (Luong and Manning 2015; Bentivogli et al. 2016). Long Short-Term Memory (Hochreiter and Schmidhuber 1997) enables NMT to conduct long-distance reordering, which is a significant challenge for Statistical Machine Translation (SMT) (Brown et al. 1993; Koehn et al. 2003).

Unlike SMT which employs a number of components, NMT adopts an end-to-end encoder-decoder framework to model the entire translation process. The role of encoder is to summarize the source sentence into a sequence of latent vectors, and the decoder acts as a language model to generate a target sentence word by word by selectively leveraging the information from the latent vectors at each step. In learning, NMT essentially estimates the likelihood of a target sentence given a source sentence.

However, conventional NMT faces two main problems:

1 Translations generated by NMT systems often lack of adequacy. When generating target words, the decoder often repeatedly selects some parts of the source sentence while ignoring other parts, which leads to over-translation and under-translation (Tu et al. 2016b). This is mainly due to that NMT does not have a mechanism to ensure that the information in the source side is completely transformed to the target side.

2 Likelihood objective is suboptimal in decoding. NMT utilizes a beam search to find a translation that maximizes the likelihood. However, we observe that likelihood favors short translations, and thus fails to distinguish good translation candidates from bad ones in a large decoding space (e.g., beam size = 100). The main reason is that likelihood only captures unidirectional dependency from source to target, which does not correlate well with translation adequacy (Li and Jurafsky 2016; Shen et al. 2016).

While previous work partially solves the above problems, in this work we propose a novel encoder-decoder-reconstructor model for NMT, aiming at alleviating these problems in a unified framework. As shown in Figure 1, given a Chinese sentence “duoge jichang beipo guanbi .”, the standard encoder-decoder translates it into an English sentence and assigns a likelihood score. Then, the newly added...
reconstructor reconstructs the translation back to the source sentence and calculates the corresponding reconstruction score. Linear interpolation of the two scores produces an overall score of the translation.

As seen, the added reconstructor imposes a constraint that an NMT model should be able to reconstruct the input source sentence from the target-side hidden layers, which encourages decoder to embed complete information of the source side. The reconstruction score serves as an auxiliary objective to measure the adequacy of translation. The combined objective consisting of likelihood and reconstruction, which measures both fluency and adequacy of translations, is used in both training and testing.

Experimental results show that the proposed approach consistently improves the translation performance when increasing the decoding space. Our model achieves a significant improvement of 2.3 BLEU points over a strong attention-based NMT system, and of 4.5 BLEU points over a state-of-the-art SMT system, trained on the same data.

Background

Encoder-Decoder based NMT

Given a source sentence $x = x_1, \ldots, x_j, \ldots, x_J$ and a target sentence $y = y_1, \ldots, y_i, \ldots, y_T$, end-to-end NMT directly models the translation probability word by word:

$$P(y|x) = \prod_{i=1}^{J} P(y_i|y_{<i}, x; \theta)$$  \hspace{1cm} (1)

where $\theta$ is the model parameters and $y_{<i} = y_1, \ldots, y_{i-1}$ is partial translation. Prediction of the $i$-th target word is generally made in an encoder-decoder framework:

$$P(y_i|y_{<i}, x; \theta) \propto \exp \left\{ f(y_{i-1}, s_i, c_i; \theta) \right\}$$  \hspace{1cm} (2)

where $s_i$ is the $i$-th hidden target state computed by the decoder Recurrent Neural Network (RNN), $c_i$ is the $i$-th source representation for generating the $i$-th target word, and $f(\cdot)$ is an activation function in the decoder. Current NMT models differ in their ways of calculating $c_i$ from the hidden states from the encoder. Please refer to (Sutskever et al. 2014; Bahdanau et al. 2015) for more details. The parameters of NMT model are trained to maximize the likelihood of a set of training examples $(x^n, y^n)_{n=1}^N$:

$$\mathcal{L}(\theta) = \arg \max_{\theta} \sum_{n=1}^{N} \log P(y^n|x^n; \theta)$$  \hspace{1cm} (3)

When generating each target word, the decoder adaptively selects partial information ($i.e., c_i$) from the encoder. This actually adopts a greedy way to select the most useful information for each generated word. There is, however, no mechanism to guarantee that the decoder conveys complete information from the source sentence to the target sentence.

In addition, we find that the performance of NMT decreases as the decoding space increases, as shown in Table 1. This is because likelihood favors short but inadequate translation candidates, which are newly added together with good candidates in larger decoding spaces. Normalizing likelihood by translation length faces the same problem.

Table: Likelihood favors short translation candidates (“Length”), and thus cannot further improve translation performance (“BLEU”) as the decoding space (“Beam”) increases. Normalizing likelihood by candidate length (“Normalization”) does not solve the problem.

<table>
<thead>
<tr>
<th>Beam</th>
<th>Likelihood</th>
<th>+ Normalization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>Length</td>
</tr>
<tr>
<td>10</td>
<td>35.46</td>
<td>29.9</td>
</tr>
<tr>
<td>100</td>
<td>25.80</td>
<td>17.9</td>
</tr>
<tr>
<td>1000</td>
<td>1.38</td>
<td>4.6</td>
</tr>
</tbody>
</table>

It is important to introduce an auxiliary objective to measure the adequacy of translation, which complements likelihood.

Reconstruction in Auto-Encoder

Reconstruction is a standard concept in auto-encoder, which is usually realized by a feed forward network (Bourlard and Kamp 1988; Vincent et al. 2010; Socher et al. 2011). The model consists of an encoding function to compute a representation from an input, and a decoding function to reconstruct the input from the representation. The parameters involved in the two functions are trained to maximize the reconstruction score, which measures the similarity between the original input and reconstructed input.

Reconstruction examines whether the reconstructed input is faithful to the original input, which is essentially similar to the consideration of adequacy in translation. It is natural to integrate reconstruction into NMT to enhance adequacy of translation. The basic idea of our approach is to reconstruct the source sentence from the latent representations of decoder, and use the reconstruction score as the adequacy measure. Analogous to auto-encoder, our approach also learns a latent representation of source sentence on the target side. Our approach can be viewed as a supervised auto-encoder in the sense that the latent representation is not only used to reconstruct the source sentence, but also used to generate the target sentence.

Approach

Architecture

We propose a novel encoder-decoder-reconstructor framework. More specifically, we base our approach on top of attention-based NMT (Bahdanau et al. 2015; Luong et al. 2015), which will be used as baseline in the experiments later. We note that the proposed approach is generally applicable to any other type of NMT architectures, such as the sequence-to-sequence model (Sutskever et al. 2014). The model architecture, shown in Figure 2, consists of two components:

- Standard encoder-decoder reads the input sentence and outputs its translation along with the likelihood score, as shown in the background section.
As shown in Figure 2, the reconstructor reads the hidden state sequence from the decoder and outputs a score of exactly reconstructing the input sentence, which we will describe below.

**Reconstructor** As shown in Figure 2, the reconstructor reconstructs the input. Here we use the hidden layer at the target side as the representation of the translation, since it plays a key role in generation of the translation. We aim at encouraging it to embed complete source information, and in the meantime to reduce the complexity of model and make the training easy.

Specifically, the reconstructor reconstructs the source sentence word by word, which is conditioned on the inverse context vector \( \hat{c}_j \), for each input word \( x_j \). The inverse context vector \( \hat{c}_j \) is computed as a weighted sum of hidden layers \( s \) at the target-side:

\[
\hat{c}_j = \sum_{i=1}^{I} \alpha_{j,i} \cdot s_i \tag{4}
\]

The weight \( \alpha_{j,i} \) of each hidden layer \( s_j \) is computed by an added inverse attention model, which has its own parameters independent from the original attention model. The reconstruction probability is calculated by

\[
R(x|s) = \prod_{j=1}^{J} R(x_j|x_{<j}, s) = \prod_{j=1}^{J} g_r(x_j-1, \hat{h}_j, \hat{c}_j) \tag{5}
\]

where \( \hat{h}_j \) is the hidden state in the reconstructor, and computed by

\[
\hat{h}_j = f_r(x_{j-1}, \hat{h}_{j-1}, \hat{c}_j) \tag{6}
\]

Here \( g_r(\cdot) \) and \( f_r(\cdot) \) are softmax function and activation function for the reconstructor, respectively. The source words \( x \) share the same word embeddings with the encoder.

**Training**

Formally, we train both the encoder-decoder \( P(y|x; \theta) \) and the reconstructor \( R(x|s; \gamma) \) on a set of training examples \( \{[x^n, y^n]\}_{n=1}^{N} \), where \( s \) is the state sequence in the decoder after generating \( y \), and \( \theta \) and \( \gamma \) are model parameters in the encoder-decoder and reconstructor respectively. The new training objective is:

\[
J(\theta, \gamma) = \arg \max_{\theta, \gamma} \sum_{n=1}^{N} \left\{ \log P(y^n|x^n; \theta) \underbrace{\log R(x^n|s^n; \gamma)}_{\text{likelihood}} + \lambda \log R(x^n|s^n; \gamma) \right\} \tag{7}
\]

where \( \lambda \) is a hyper-parameter that balances the preference between likelihood and reconstruction.

Note that the objective consists of two parts: likelihood measures translation fluency, and reconstruction measures translation adequacy. It is clear that the combined objective is more consistent with the goal of enhancing overall translation quality, and can more effectively guide the parameter training for making better translation.

**Testing**

Once a model is trained, we use a beam search to find a translation that approximately maximizes both the likelihood and reconstruction score. As shown in Figure 3, given an input sentence, a two-phase scheme is used:

1. The standard encoder-decoder produces a set of translation candidates, each of which is a triple consisting of a translation candidate, its corresponding hidden layer at the target-side \( s \), and its likelihood score \( P \).

![Figure 2: Architecture of NMT with reconstruction, which introduces a reconstructor to map from the hidden layer at the target side to the original input.](image)

![Figure 3: Illustration of testing with reconstructor.](image)
For each translation candidate, the reconstructor reads its corresponding hidden layer at the target-side and outputs an auxiliary reconstruction score $R$. Linear interpolation of likelihood $P$ and reconstruction score $R$ produces an overall score, which is used to select the final translation.

In testing, reconstruction works as a reranking technique to select a better translation from the $k$-best candidates generated by the decoder.

**Experiments**

**Setup**

We carry out experiments on Chinese-English translation. The training dataset consists of 1.25M sentence pairs extracted from LDC corpora, with 27.9M Chinese words and 34.5M English words respectively. We choose the NIST 2002 (MT02) dataset as validation set, and the NIST 2005 (MT05), 2006 (MT06) and 2008 (MT08) datasets as test sets. We use the case-insensitive 4-gram NIST BLEU score (Papineni et al. 2002) as evaluation metric, and sign-test (Collins et al. 2005) for statistical significance test.

We compare our method with state-of-the-art SMT and NMT models:

- **MOSES** (Koehn et al. 2007): an open source phrase-based translation system with default configuration and a 4-gram language model trained on the target portion of training data.

- **RNNSEARCH**: our re-implemented attention-based NMT system, which incorporates dropout (Hinton et al. 2012) on the output layer and improves the attention model by feeding the lastly generated word.

For training RNNSEARCH, we limit the source and target vocabularies to the most frequent 30K words in Chinese and English. We train each model with the sentences of length up to 80 words in the training data. We shuffle mini-batches as we proceed and the mini-batch size is 80. The word embedding dimension is 620 and the hidden layer dimension is 1000. We train for 15 epochs using Adadelta (Zeiler 2012).

For our model, we use the same setting as RNNSEARCH if applicable. We set the hyper-parameter $\lambda = 1$. The parameters of our model (i.e., encoder and decoder, except those related to reconstructor) are initialized by the RNNSEARCH model trained on a parallel corpus. We further train all the parameters of our model for another 10 epochs.

**Correlation between Reconstruction and Adequacy**

<table>
<thead>
<tr>
<th>Evaluator</th>
<th>Adequacy</th>
<th>Fluency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluator1</td>
<td>0.514</td>
<td>0.301</td>
</tr>
<tr>
<td>Evaluator2</td>
<td>0.499</td>
<td>0.307</td>
</tr>
</tbody>
</table>

Table 2: Correlation between reconstruction score and translation adequacy (and fluency).

In the first experiment, we investigate the validity of our assumption that reconstruction score correlates well with translation adequacy, which is the underlying assumption of the approach. We conduct a subjective evaluation: two human evaluators are asked to evaluate the translations of 200 source sentences randomly sampled from the test sets. We calculate Pearson Correlation between the reconstruction scores and the corresponding adequacy and fluency scores on the samples, as shown in Table 2. Two evaluators produce similar results: reconstruction score is more related to translation adequacy than fluency.

**Effect of Reconstruction on Translation**

![Figure 4: Learning curves – translation (left y-axis) and reconstruction (right y-axis) performances (in BLEU scores) on the validation set as training progresses. We find our approach is capable of generating better translations over time by better reconstructing original inputs.](image)

In this experiment, we investigate the effect of reconstruction on translation performance over time, which is measured in BLEU scores on the validation set. For reconstruction, we use the reconstructor to stochastically generate a source sentence for each translation, and calculate the BLEU score of the reconstructed input under the reference of the original input. Generally, as shown in Figure 4, the BLEU score of translation goes up with the improvement of reconstruction over time. The translation performance reaches a peak at iteration 110K, when the model achieves a balance between likelihood and reconstruction score. Therefore, we use the trained model at iteration 110K in the following experiments.

**Effect of Reconstruction in Large Decoding Space**

Can our approach cope with the limitation of likelihood in large decoding spaces? To answer this question, we investigate the effect of reconstruction on different beam sizes $k$, as shown in Table 3. Our approach can indeed solve the problem: increasing the size of decoding space generally leads to improving the BLEU score. We attribute this to the ability...
Andrea Core, in the following experiments we set the max value of accuracy but greatly decreases decoding efficiency. There-

does not result in significant improvements of translation between
equacy of translation candidates. There is a significant gap
of the combined objective to measure both fluency and ad-

Table 3: Translation performances of different decoding
beams on the validation set, in which the averaged length
of reference translations is 27.0.

of the combined objective to measure both fluency and ade-
quacy of translation candidates. There is a significant gap
between
k = 10 and k = 100. However, keeping increasing
k does not result in significant improvements of translation
accuracy but greatly decreases decoding efficiency. There-
fore, in the following experiments we set the max value of
k to 100, and use normalized likelihood for k = 100 if we
don’t use reconstruction in testing.

Main Results

Table 4 shows the translation performances on test sets mea-
sured in BLEU score. RNNSearch significantly outper-
forms Moses by 2.2 BLEU points on average, indicating
that it is a strong baseline NMT system. This is mainly due
to the introduction of two advanced techniques. Increasing
beam size leads to decreasing translation performances on
test sets, which is consistent with the result on the vali-
dation set. We compare our methods with “RNNSearch
(Bundle=10)” in the following analysis, since it yields the best
performance in the baseline systems.

First, the introduction of reconstruction significantly im-
proves the performance over baseline by 1.1 BLEU points
with beam size k = 10. Most importantly, we obtain a fur-
ther improvement of 1.2 BLEU points when expanding the
decoding space. Second, our approach also consistently im-
proves the quality (in terms of Oracle score, see the last
column) of k-best translation candidates over the baseline
system on various beam sizes. This confirms our claim that
the combined objective contributes to parameter training for
generating better translation candidates.

Analysis

We conduct extensive analyses to better understand our
model in terms of efficiency of the added reconstruction,
 contributed of reconstruction from training and testing, al-
leviating typical translation problems, and building the abil-
ity of handling long sentences.

Table 4: Evaluation of translation quality. “Oracle” is the best BLEU score that the k-best translation candidates can achieve on
all the test sets. “*” indicate statistically significant difference (p < 0.01) from the baseline “RNNSearch (Bundle=10)”.

<table>
<thead>
<tr>
<th>Model</th>
<th>Beam</th>
<th>Tuning</th>
<th>MT05</th>
<th>MT06</th>
<th>MT08</th>
<th>All</th>
<th>Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moses</td>
<td>100</td>
<td>34.03</td>
<td>31.37</td>
<td>30.85</td>
<td>23.01</td>
<td>28.44</td>
<td>35.17</td>
</tr>
<tr>
<td>RNNSearch</td>
<td>10</td>
<td>35.46</td>
<td>32.63</td>
<td>32.85</td>
<td>25.96</td>
<td>30.65</td>
<td>34.27</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>33.39</td>
<td>29.58</td>
<td>30.21</td>
<td>23.76</td>
<td>27.97</td>
<td>40.20</td>
</tr>
<tr>
<td>RNNSearch+Reconstruction</td>
<td>10</td>
<td>36.34*</td>
<td>33.73</td>
<td>34.15*</td>
<td>26.85*</td>
<td>31.73*</td>
<td>36.05</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>37.35*</td>
<td>34.88</td>
<td>35.19*</td>
<td>27.93*</td>
<td>32.94*</td>
<td>42.49</td>
</tr>
</tbody>
</table>

Table 3: Translation performances of different decoding
beam sizes on the validation set, in which the averaged length
of reference translations is 27.0.

Table 5: Contributions of reconstruction from parameter
training and reranking of candidates in testing.

<table>
<thead>
<tr>
<th>Beam</th>
<th>Likelihood</th>
<th>+Reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>Length</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>Length</td>
</tr>
<tr>
<td>10</td>
<td>35.46</td>
<td>29.9</td>
</tr>
<tr>
<td></td>
<td>36.34</td>
<td>29.1</td>
</tr>
<tr>
<td>100</td>
<td>25.80</td>
<td>17.9</td>
</tr>
<tr>
<td></td>
<td>37.35</td>
<td>27.1</td>
</tr>
<tr>
<td>1000</td>
<td>1.38</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td>37.77</td>
<td>26.6</td>
</tr>
</tbody>
</table>

Speed

Introducing reconstruction significantly slows
the decoding speed, while it slightly decreases the
decoding speed. For training, when running on a single GPU
device Tesla K80, the speed of the baseline model is 960
target words per second, while the speed of the proposed
model is 500 target words per second. For decoding with
beam=10, the speed of the baseline model is 2.28 seconds
per sentence, while that of the proposed approach is 2.60
seconds per sentence. We attribute the effectiveness of
decoding to the avoidance of beam search for reconstruction
and the benefit of batch computation on GPU.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNSearch</td>
<td>18.2%</td>
<td>3.9%</td>
</tr>
<tr>
<td>RNNSearch+Rec.</td>
<td>16.2%</td>
<td>2.4%</td>
</tr>
</tbody>
</table>

Table 6: Subjective evaluation of translation problems.
Numbers denote percentages of source words.

Problem Analysis

We then conduct a subjective evaluation to investigate the benefit of incorporating reconstruction on the randomly selected 200 sentences. Table 6 shows the results of subjective evaluation on translation. RNNSearch suffers from serious under-translation and

4For decoding with beam=100, the speeds are 22.97 and 25.29
seconds per sentence, respectively.
over-translation problems, which is consistent with the finding in other work (Tu et al. 2016b). Incorporating reconstruction significantly alleviates these problems, and reduces 11.0% and 38.5% of under-translation and over-translation errors respectively. The main reason is that both under-translation and over-translation lead to lower reconstruction scores, and thus are penalized by the reconstruction objective. As a result, the corresponding candidate is less likely to be selected as the final translation.

<table>
<thead>
<tr>
<th>Model</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNSEARCH</td>
<td>30.65</td>
</tr>
<tr>
<td>RNNSEARCH+Cov.</td>
<td>31.89</td>
</tr>
<tr>
<td>RNNSEARCH+Cov.+Rec.</td>
<td>33.44</td>
</tr>
<tr>
<td>RNNSEARCH+Ctx.</td>
<td>32.05</td>
</tr>
<tr>
<td>RNNSEARCH+Ctx.+Rec.</td>
<td>33.51</td>
</tr>
<tr>
<td>RNNSEARCH+Cov.+Ctx.</td>
<td>33.12</td>
</tr>
<tr>
<td>RNNSEARCH+Cov.+Ctx.+Rec.</td>
<td>34.09</td>
</tr>
</tbody>
</table>

Table 7: Comparison with previous work on enhancing adequacy of NMT. “Cov.” denotes coverage mechanism to keep track of the attention history (Tu et al. 2016b), and “Ctx.” denotes context gate to dynamically control the ratios at which source and target contexts contribute to the generation of target words (Tu et al. 2016a).

Enhancing Translation Adequacy Recently, several work shows that NMT favors fluent but inadequate translations (Tu et al. 2016b; 2016a). While all the work is towards enhancing adequacy of NMT, our approach is complimentary: the above work is still under the standard encoder-decoder framework, while we propose a novel encoder-decoder-reconstructor framework. Experiments show that combining those models together can further improve the translation performance.

Improving Beam Search Standard NMT models exploit a simple beam search algorithm to generate the translation word by word. Several researchers rescore word candidates with additional features, such as language model probability (Gulcehre et al. 2015) and SMT features (He et al. 2016; Stahlberg et al. 2016). In contrast, Li and Jurafsky (2016) rescore translation candidates on sentence-level with the mutual information between source and target sides. In the above work, NMT is treated as a black-box and its weighted outputs are combined with other features only in testing. In this work, we move forward further by incorporating reconstruction score into the objective of training, which leads to creation of better translation candidates.

Capturing Bidirectional Dependency Standard NMT models only capture the unidirectional dependency from source to target with the likelihood objective. It has been shown that combination of two directional models outperforms each model alone (Liang et al. 2006; Cheng et al. 2016a; Cheng et al. 2016b). Among them, Cheng et al. (2016b) reconstruct the monolingual corpora with two separate source-to-target and target-to-source NMT models. Closely related to Cheng et al. (2016b), our approach aims at enhancing adequacy of unidirectional (i.e., source-to-target) NMT via an auxiliary target-to-source objective on parallel corpora, while theirs focuses on learning bidirectional NMT models via auto-encoders on monolingual corpora. Therefore, we use the decoder states as the input of the reconstructor, to encourage the target representation to contain the complete source information to reconstruct back to the source sentence.

Figure 5: Performance of the generated translations with respect to the lengths of the input sentences on the test sets.

Length Analysis Following Bahdanau et al. (2015), we group sentences of similar lengths together and compute the BLEU score for each group, as shown in Figure 5. Clearly the proposed approach outperforms all the other systems in all length segments. Specifically, RNNSEARCH outperforms Moses on all sentence segments, while its performance degrades faster than its competitors, which is consistent with the finding in (Bentivogli et al. 2016). This is mainly due to that RNNSEARCH seriously suffers from inadequate translations on long sentences (Tu et al. 2016b). Our model explicitly encourages the decoder to incorporate source information as much as possible, and thus the improvements are more significant on long sentences.

Comparison with Previous Work We re-implement the methods of Tu et al. (2016b; 2016a) on top of RNNSEARCH. For the coverage mechanism (Tu et al. 2016b), we use the neural network based coverage, and the coverage dimension is 100. For the context gates (Tu et al. 2016a), we apply them on both source and target sides. Table 7 lists the comparison results. Coverage mechanism and context gates significantly improve translation performance individually, and combining them achieves a further improvement. This is consistent with the results in (Tu et al. 2016b; 2016a). Our model consistently improves the translation performance when further combined with the models.

Related Work

Our work is inspired by research on improving NMT by:

- Enhancing Translation Adequacy
- Improving Beam Search
- Capturing Bidirectional Dependency
Conclusion
We propose a novel encoder-decoder-reconstructor framework for NMT, in which the newly added reconstructor introduces an auxiliary score to measure the adequacy of translation candidates. The advantage of the proposed approach is of two-fold. First, it improves parameter training for producing better translation candidates. Second, it consistently improves translation performance when the decoding space increases, while conventional NMT fails to do so. Experimental results show that the two advantages can indeed help our approach to consistently improve translation performance.

There is still a significant gap between de facto translation and oracle of k-best translation candidates, especially when the decoding space increases. We plan to narrow the gap with rich features, which can better measure the quality of translation candidates. It is also necessary to validate the effectiveness of our approach on more language pairs and other NMT architectures.

Acknowledgement
This work is supported by China National 973 project 2014CB340301. Yang Liu is supported by the National Natural Science Foundation of China (No. 61522204) and the 863 Program (2015AA015407).

References