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:

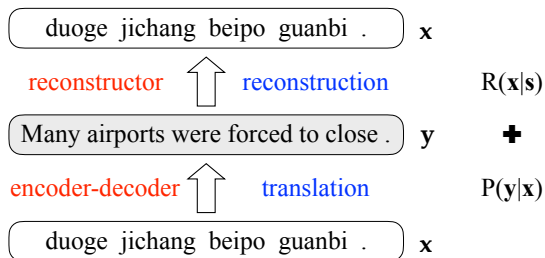1 *Translations generated by NMT systems often lack of adequacy*. When generating target words, the decoder often

Figure 1: Example of NMT with reconstruction. Our idea is to leverage reconstruction score $R(\mathbf{x}|\mathbf{s})$ as an auxiliary objective to measure the adequacy of translation candidate, where $\mathbf{s}$ is the target-side hidden layer in decoder for generating the translation $\mathbf{y}$. Linear interpolation of likelihood score $P(\mathbf{y}|\mathbf{x})$ and reconstruction score is used to (1) improve parameter learning for generating better translation candidates in *training*, and (2) conduct better rerank of generated candidates in *testing*.

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

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{I} P(y_i|y_{<i}, \mathbf{x}; \theta) \qquad (1)$$

where θ 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:

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

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 $\{[\mathbf{x}^n, \mathbf{y}^n]\}_{n=1}^{N}$:

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

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

Beam	Likelihood		+ Normalization	
	BLEU	Length	BLEU	Length
10	35.46	29.9	34.51	32.3
100	25.80	17.9	33.39	32.7
1000	1.38	4.6	29.50	33.9

Table 1: 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.

candidates in larger decoding spaces. Normalizing likelihood by translation length faces the same problem.

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 prepose 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.
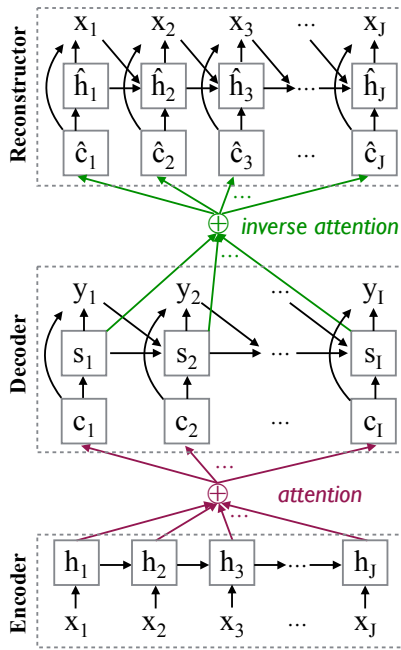
Figure 2: Architecture of NMT with reconstruction, which introduces a reconstructor to map from the hidden layer at the target side to the original input.

- Added *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 $\mathbf{s}$ at the target-side:

$$\hat{c}_j = \sum_{i=1}^{I} \hat{\alpha}_{j,i} \cdot s_i \tag{4}$$

The weight $\hat{\alpha}_{j,i}$ 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

$$
\begin{aligned}
R(\mathbf{x}|\mathbf{s}) &= \prod_{j=1}^{J} R(x_j|x_{<j}, \mathbf{s}) \\
&= \prod_{j=1}^{J} g_r(x_{j-1}, \hat{h}_j, \hat{c}_j) \tag{5}
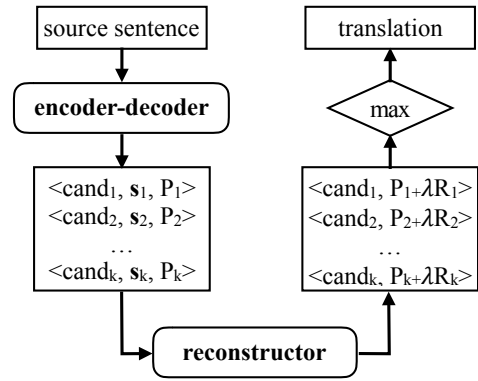\end{aligned}
$$

Figure 3: Illustration of testing with reconstructor.

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 $\mathbf{x}$ share the same word embeddings with the encoder.

Training

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

$$
J(\theta, \gamma) = \arg\max_{\theta, \gamma} \sum_{n=1}^{N} \Bigg\{ \underbrace{\log P(\mathbf{y}^n|\mathbf{x}^n; \theta)}_{likelihood} \\
+ \lambda \underbrace{\log R(\mathbf{x}^n|\mathbf{s}^n; \gamma)}_{reconstruction} \Bigg\} \tag{7}
$$

where λ 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 $\mathbf{s}$, and its likelihood score P.

2 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.[1]

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.[2] 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

	Adequacy	Fluency
Evaluator1	0.514	0.301
Evaluator2	0.499	0.307

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

---

[1]Interpolation weight λ in testing is the same as in training.

[2]The corpora include LDC2002E18, LDC2003E07, LDC2003E14, LDC2004T07, LDC2004T08 and LDC2005T06.

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.
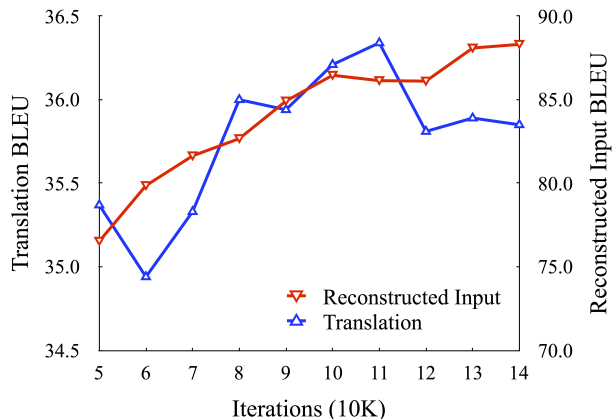
Effect of Reconstruction on Translation

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.

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,[3] 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

---

[3]Note that it is different from the standard procedure, which calculates the probability of exactly reconstructing the original input.

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

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 (Beam=10)".

Beam	Likelihood		+Reconstruction	
	BLEU	Length	BLEU	Length
10	35.46	29.9	36.34	29.1
100	25.80	17.9	37.35	27.1
1000	1.38	4.6	37.77	26.6

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 adequacy 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. Therefore, 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 measured in BLEU score. RNNSEARCH significantly outperforms 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 validation set. We compare our methods with "RNNSEARCH (Beam=10)" in the following analysis, since it yields the best performance in the baseline systems.

First, the introduction of reconstruction significantly improves the performance over baseline by 1.1 BLEU points with beam size $k = 10$. Most importantly, we obtain a further improvement of 1.2 BLEU points when expanding the decoding space. Second, our approach also consistently improves 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, contribution of reconstruction from training and testing, alleviating typical translation problems, and building the ability of handling long sentences.

Speed Introducing reconstruction significantly slows down the training 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.[4] We attribute the effectiveness of decoding to the avoidance of beam search for reconstruction and the benefit of batch computation on GPU.

Rec. used in		Beam	
Training	*Testing*	10	100
×	×	30.65	27.97
✓	×	31.17	31.51
✓	✓	31.73	32.94

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

Contribution Analysis The contribution of reconstruction is of two-fold: (1) enabling parameter training for generating better translation candidates, and (2) enabling better reranking of generated candidates in testing. Table 5 lists the improvements from the two contribution sources. When applied only in training, reconstruction improves translation performance by generating fluent and adequate translation candidates. On top of that, reconstruction-based reranking further improves the performance. The improvements are more significant when decoding spaces increase.

Model	Under-Tran.	Over-Tran.
RNNSEARCH	18.2%	3.9%
RNNSEARCH+Rec.	16.2%	2.4%

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

---

[4]For 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.
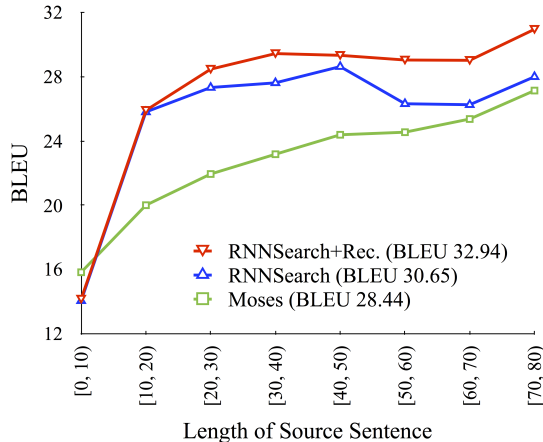

Figure 5: Performance of the generated translations with respect to the lengths of the input sentences on the test sets.

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:

Model	Test	$\triangle$
RNNSEARCH	30.65	
RNNSEARCH+Cov.	31.89	
RNNSEARCH+Cov.+*Rec.*	33.44	+1.6
RNNSEARCH+Ctx.	32.05	
RNNSEARCH+Ctx.+*Rec.*	33.51	+1.5
RNNSEARCH+Cov.+Ctx.	33.12	
RNNSEARCH+Cov.+Ctx.+*Rec.*	34.09	+1.0

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.

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

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Bentivogli, L.; Bisazza, A.; Cettolo, M.; and Federico, M. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *EMNLP 2016*.

Bourlard, H., and Kamp, Y. 1988. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics* 59(4-5):291–294.

Brown, P. E.; Pietra, S. A. D.; Pietra, V. J. D.; and Mercer, R. L. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19(2):263–311.

Cheng, Y.; Shen, S.; He, Z.; He, W.; Wu, H.; Sun, M.; and Liu, Y. 2016a. Agreement-based joint training for bidirectional attention-based neural machine translation. In *IJCAI 2016*.

Cheng, Y.; Xu, W.; He, Z.; He, W.; Wu, H.; Sun, M.; and Liu, Y. 2016b. Semi-Supervised Learning for Neural Machine Translation. In *ACL 2016*.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP 2014*.

Collins, M.; Koehn, P.; and Kučerová, I. 2005. Clause restructuring for statistical machine translation. In *ACL 2005*.

Gulcehre, C.; Firat, O.; Xu, K.; Cho, K.; Barrault, L.; Lin, H.-C.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2015. On Using Monolingual Corpora in Neural Machine Translation. *arXiv*.

He, W.; He, Z.; Wu, H.; and Wang, H. 2016. Improved neural machine translation with smt features. In *AAAI 2016*.

Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*.

Kalchbrenner, N., and Blunsom, P. 2013. Recurrent continuous translation models. In *EMNLP 2013*.

Koehn, P.; Och, F. J.; and Marcu, D. 2003. Statistical phrase-based translation. In *NAACL 2003*.

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; and Herbst, E. 2007. Moses: open source toolkit for statistical machine translation. In *ACL 2007*.

Li, J., and Jurafsky, D. 2016. Mutual information and diverse decoding improve neural machine translation. In *NAACL 2016*.

Liang, P.; Taskar, B.; and Klein, D. 2006. Alignment by agreement. In *NAACL 2006*.

Luong, M.-T., and Manning, C. D. 2015. Stanford neural machine translation systems for spoken language domains. In *IWSLT 2015*.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP 2015*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*.

Shen, S.; Cheng, Y.; He, Z.; He, W.; Wu, H.; Sun, M.; and Liu, Y. 2016. Minimum Risk Training for Neural Machine Translation. In *ACL 2016*.

Socher, R.; Pennington, J.; Huang, E. H.; Ng, A. Y.; and Manning, C. D. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP 2011*.

Stahlberg, F.; Hasler, E.; Waite, A.; and Byrne, B. 2016. Syntactically Guided Neural Machine Translation. *arXiv*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS 2014*.

Tu, Z.; Liu, Y.; Lu, Z.; Liu, X.; and Li, H. 2016a. Context Gates for Neural Machine Translation. In *arXiv*.

Tu, Z.; Lu, Z.; Liu, Y.; Liu, X.; and Li, H. 2016b. Modeling Coverage for Neural Machine Translation. In *ACL 2016*.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11(Dec):3371–3408.

Zeiler, M. D. 2012. ADADELTA: an adaptive learning rate method. *arXiv*.