Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Improved Neural Machine Translation with SMT Features

Wei He, Zhongjun He,* Hua Wu, and **Haifeng Wang**
Baidu Inc. No. 10, Shangdi 10th Street, Beijing, 100085, China
{hewei06, hezhongjun, wu_hua, wanghaifeng}@baidu.com

Abstract

Neural machine translation (NMT) conducts end-to-end translation with a source language encoder and a target language decoder, making promising translation performance. However, as a newly emerged approach, the method has some limitations. An NMT system usually has to apply a vocabulary of certain size to avoid the time-consuming training and decoding, thus it causes a serious out-of-vocabulary problem. Furthermore, the decoder lacks a mechanism to guarantee all the source words to be translated and usually favors short translations, resulting in fluent but inadequate translations. In order to solve the above problems, we incorporate statistical machine translation (SMT) features, such as a translation model and an n-gram language model, with the NMT model under the log-linear framework. Our experiments show that the proposed method significantly improves the translation quality of the state-of-the-art NMT system on Chinese-to-English translation tasks. Our method produces a gain of up to 2.33 BLEU score on NIST open test sets.

Introduction

Neural networks have recently been applied to machine translation and begun to show promising results. Sutskever, Vinyals, and Le (2014) and Bahdanau, Cho, and Bengio (2014) directly built neural networks to perform end-to-end translation, named neural machine translation (NMT). Typically, an NMT system contains two components, an encoder that converts a source sentence into a vector, and a decoder that generates target translation based on the vector.

The strength of NMT lies in that the semantic and structural information can be learned by taking global context into consideration. However, as a newly emerged approach, the NMT method has some limitations that may jeopardize its ability to generate better translation.

1. To reduce model complexity, an NMT system usually uses the top-N frequent words in the training corpus and regards other words as unseen ones, which causes a serious out-of-vocabulary (OOV) problem. When OOV words occur in the sentences to be translated, the translation quality would be badly hurt.
2. The NMT decoder lacks a mechanism to guarantee that all the source words are translated and usually favors short translations. This sometimes results in an inadequate translation that does not convey the complete meaning of source sentence.
3. NMT models cannot make use of large amount of target monolingual corpus. Therefore, it is difficult for an NMT system to benefit from target language model trained on target monolingual corpus, which is proven to be useful for improving translation quality in statistical machine translation (SMT).

Luong et al. (2015) used a dictionary to translate the OOV words in a post-processing step. Gulcehre et al. (2015) proposed two ways to integrate a recurrent neural network (RNN) based language model into the NMT model. However, these methods only focus on one of the above NMT problems.

Intuitively, these problems could be alleviated with some of the SMT components, such as the translation table, the n-gram language model. Nevertheless, the current NMT framework suffers from a fact that it is difficult to add effective features into the model to further improve translation quality.

In this paper, we propose to improve NMT by integrating SMT features with the NMT model under the log-linear framework. We incorporate 3 SMT features, including the translation model, the word reward feature and the n-gram language model. The translation model is trained on word-aligned bilingual corpus with the conventional phrase-based SMT approach (Koehn, Och, and Marcu 2003), and employed to score word pairs and alleviate the OOV problem. The word reward feature controls the length of the translation. And the n-gram language model aims to enhance the local fluency which is trained on target monolingual sentences.

Compared to previous methods, our method has the following advantages:

1. The log-linear framework makes an NMT system be easily extended. It can be integrated with effective features used in conventional SMT models.
2. We integrate a word translation table into the log-linear framework with the translation probabilities estimated from the word-aligned bilingual corpus which is trained

*Corresponding author: Zhongjun He hezhongjun@baidu.com.

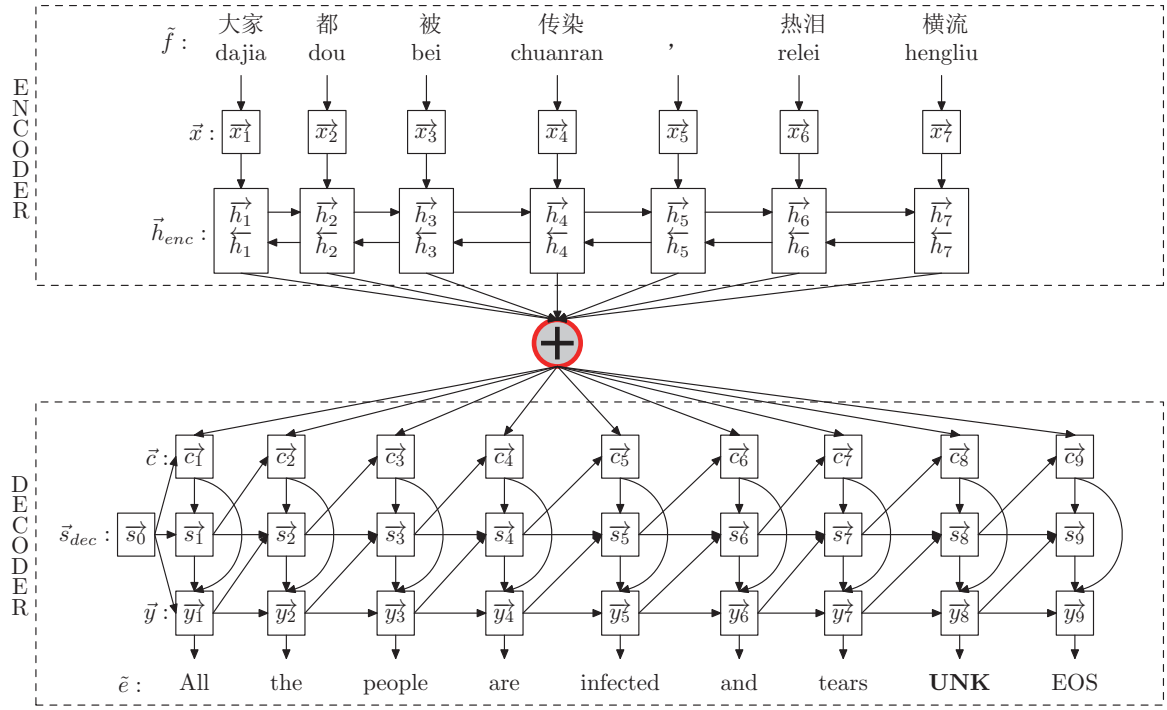

Figure 1: Illustration of the RNN Encoder-Decoder. Note that there is a UNK symbol on the target side because of the OOV problem.

on IBM models (Brown et al. 1993). The translation table can not only be used to recover the translation of such words that are taken as unknown words in the NMT system, but also provides another way to measure the correlation between source and target words. The translation table is integrated into the decoding procedure rather than used in a post-process step (Luong et al. 2015).

3. To alleviate the inadequate translation problem in NMT, we use a word reward feature to make the decoder favor long translation, rather than using a coverage vector that guarantees all source words are translated.
4. We add an n-gram language model to the log-linear framework, so as to make full use of the large-scale monolingual corpus to further improve translation quality. The language model score along with other features is used to select candidates during decoding. While in the shallow fusion, Gulcehre et al. (2015) used the language model to rescore top-N candidates generated by the NMT decoder.

Our approach is different from the conventional methods which integrated neural networks into SMT systems (Devlin et al. 2014; Auli et al. 2013; Cho et al. 2014; Li, Liu, and Sun 2013; Zhai et al. 2013). The main difference is that the conventional methods are conducted within an SMT framework. Our system is an NMT system, enhanced by effective SMT features.

We carried out experiments with an open-source NMT system GroundHog [1] (Bahdanau, Cho, and Bengio 2014). The system builds two RNNs to perform end-to-end translation: one as an encoder and the other as a decoder. We trained the system with a large amount corpus (containing about 200 million sentence pairs) collected from the web. Experiments on Chinese-to-English translation tasks demonstrate that the proposed method achieves significant improvements over the state-of-the-art NMT system.

[1] https://github.com/lisa-groundhog/GroundHog

Background

This section briefly reviews the RNN encoder-decoder, a recently proposed NMT approach based on recurrent neural network, and the log-linear models, the dominant framework for SMT in the last decade.

RNN Encoder-Decoder

Figure 1 shows the translation procedure of the RNN encoder-decoder (Bahdanau, Cho, and Bengio 2014) for Chinese-to-English translation. Given a source sentence $\tilde{f} = f_1, f_2, ..., f_I$, the encoder first encodes $\tilde{f}$ into a sequence of vectors, then the decoder generates the target translation $\tilde{e} = e_1, e_2, ..., e_J$ based on the vectors and the target words previously generated.

The encoder is a bidirectional RNN (Schuster and Paliwal 1997) with a hidden layer. At the encoding step, the encoder firstly projects the input sentence $\tilde{f}$ into word vectors $\vec{x} = (x_1, x_2, ..., x_I), x_i \in \mathbb{R}^{K_x}$, where K_x is the vocabulary size of the source language. Then the network updates the hidden state h^i_{enc} at each step by

$$h^i_{enc} = g_{enc}(x_i, h^{i-1}_{enc}) \quad (1)$$

where, g_{enc} is an activation function, e.g. the $tanh$ function. $h^i_{enc} = [\overrightarrow{h^i_{enc}}^\top, \overleftarrow{h^i_{enc}}^\top]^\top$ is the concatenation of the forward and backward hidden states calculated based on the source sentence.

At the decoding step, the probability of the output sequence is computed as:

$$p(\vec{y}) = \prod_{j=1}^{J} p(y_j|\{y_{j-1}, y_{j-2}, ..., y_1\}, \vec{x}) \quad (2)$$

$$= \prod_{j=1}^{J} g_{dec}(s^j_{dec}, y_{j-1}, c_j) \quad (3)$$

where, s^j_{dec} is the hidden state at step j, which is computed by,

$$s^j_{dec} = g'_{dec}(s^{j-1}_{dec}, y_{j-1}, c_j) \quad (4)$$

g_{dec} and g'_{dec} are non-linear activation functions. The context vector c_j is computed as a weighted sum of the hidden states of the encoder:

$$c_j = \sum_{i=1}^{T_x} \alpha_{ji} h^i_{enc} \quad (5)$$

where, the weight α_{ji} can be considered as an association measure that how well a target word y_j is translated from a source word x_i. Bahdanau, Cho, and Bengio (2014) used a feed-forward neural network to parametrize an alignment model to estimate α. This is an important difference from the basic RNN encoder-decoder proposed by Cho et al. (2014), which encodes the source sentence into a single vector with fixed length, thus unable to reflect the strength of the relationship between source and target words.

Following Cho et al. (2014), Bahdanau, Cho, and Bengio (2014) also used two types of hidden units, *reset* gates and *update* gates. The *reset* gates allow the networks to ignore the information of some previous hidden states, which may be noisy for the current state. The *update* gates control the degree of the information being transfered to the current state from the previous states. The role of the two kinds of gates is analogous to the long-short-term-memory (LSTM) (Sutskever, Vinyals, and Le 2014), but much simpler.

The RNN encoder-decoder is trained on bilingual corpora and performs an end-to-end translation. However, under the current architecture, it is difficult to improve the translation quality by integrating additional translation knowledge.

Log-linear Models

The widely used log-linear framework in SMT was introduced by Och and Ney (2002).

$$p(\tilde{e}|\tilde{f}) = \frac{exp(\sum_{i=1}^{m} \lambda_i H_i(\tilde{f}, \tilde{e}))}{\sum_{\tilde{e}'} exp(\sum_{i=1}^{m} \lambda_i H_i(\tilde{f}, \tilde{e}'))} \quad (6)$$

where, $H_i(\tilde{f}, \tilde{e})$ is a feature function and λ_i is the weight.

The strength of the log-linear model is that features can be easily added into it. A standard phrase-based SMT (Koehn, Och, and Marcu 2003) typically contains 8 features: the bi-directional translation probabilities $p(f|e)$ and $p(e|f)$, the bi-directional lexical weights $p_{lex}(f|e)$ and $p_{lex}(e|f)$, the language model, the reordering model, the word penalty, and the phrase penalty. These features have been proven effective to improve translation quality.

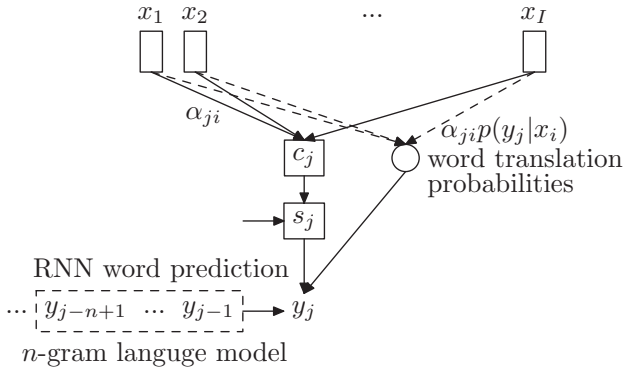

Figure 2: Illustration of the Log-linear NMT. To predict the target word y_j, we introduce SMT features, such as the word translation probabilities, the n-gram language model, together with the probabilities estimated by RNNs.

Recently, some researchers integrated neural networks as a component into SMT systems, to improve language modeling (Devlin et al. 2014), translation modeling (Auli et al. 2013; Cho et al. 2014), and phrase reordering (Li, Liu, and Sun 2013; Zhai et al. 2013).

The important difference between our method and the previous methods is that we instead integrate SMT features with the NMT model via the log-linear framework, making the NMT model extendable.

Log-linear NMT

We believe that integrating SMT features might help improve translation quality for the NMT systems. We use Figure 2 to illustrate our idea. At each step to predict a target word y_j, in addition to the probabilities estimated by RNN, we add a word translation table and an n-gram language model. The translation table, estimating from word-aligned bilingual corpus, can improve lexical translation and translate the low-frequency words which are taken as unknown words. The language model can make full use of target monolingual corpus to improve local fluency. We use a log-linear framework to integrate these effective features.

Feature Definition

Our method includes the following feature functions:

1. The RNN encoder-decoder feature. This feature is the conditional probability estimated by the NMT model that predicts a target word based on the source sentence and previously produced target words.

$$H_{rnn} = \sum_{j=1}^{J} log(g(y_{j-1}, s_j, c_j)) \quad (7)$$

2. The bi-directional word translation probabilities. At each step of decoding, we estimate the lexical translation prob-

abilities between target candidates and the corresponding source words.

$$H_{tp1} = \sum_{j=1}^{J}\sum_{i=1}^{I} \alpha_{ji} log(p(y_j|x_i)) \quad (8)$$

$$H_{tp2} = \sum_{j=1}^{J}\sum_{i=1}^{I} \alpha_{ji} log(p(x_i|y_j)) \quad (9)$$

where, α_{ji} is the weighted soft alignments between the target word y_j and associated source words, estimated by the RNN encode-decoder (Section *RNN Encoder-Decoder*). $p(y|x)$ and $p(x|y)$ is the word translation probabilities estimated from word-aligned bilingual corpus, where the word alignment is trained with GIZA++ (Och and Ney 2004) and the "grow-diag-final" (Koehn, Och, and Marcu 2003) method.

The word translation probabilities are computed as follows:

$$p(x|y) = \frac{N(x,y)}{\sum_{x'} N(x',y)} \quad (10)$$

$$p(y|x) = \frac{N(y,x)}{\sum_{y'} N(y',x)} \quad (11)$$

where $N(x,y)$ is the co-occurrence of the corresponding words x and y.

3. The standard n-gram language model.

$$H_{lm} = \sum_{j=1}^{J} log(p(y_j|y_{j-1},...,y_{j-n+1})) \quad (12)$$

The language model is trained on target monolingual corpus. Thus this feature allows us to make use of a large-scale monolingual corpus of target language.

4. The word reward feature.

$$H_{wp} = \sum_{j=1}^{J} 1 \quad (13)$$

The feature is the number of words in the target sentence, which could control an appropriate length of translation.

Compared with the original RNN model, we add three additional features for each state generated by the RNN decoder during decoding. The translation can be generated from the final state with the highest total score.

Handling the OOV Problem

As mentioned, the NMT encoder-decoder usually faces a serious OOV problem. The post-processing method (Luong et al. 2015) did not benefit from the contextual information during decoding. We instead use a word translation table, automatically extracted from word-aligned bilingual corpus, to translate the OOV words during decoding.

See Figure 3 for illustration. In order to produce the correct translation for the OOV word, we firstly find its corresponding source word. According to the alignment probabilities estimated by the RNN model, the "UNK" symbol

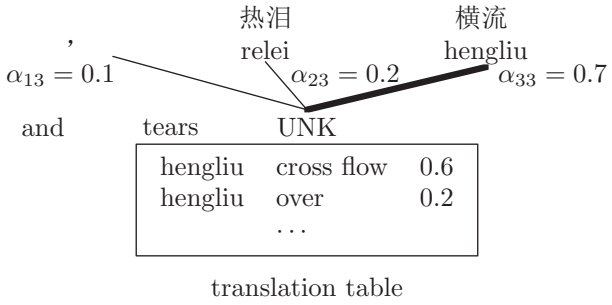

Figure 3: Illustration for recovering unknown word using translation table. α_{ij} is the alignment probabilities estimated by the RNN encoder-decoder.

refers to the source word "*hengliu*". Then we obtain translation candidates from the word translation table. The final translation is determined by the proposed log-linear model, considering rich contextual information on both the source and target sides during decoding.

Decoding

The RNN decoder utilizes a beam search algorithm (Bahdanau, Cho, and Bengio 2014) to produce translation that maximizes the conditional translation probability,

$$\hat{y} = \arg\max_{\tilde{y}} p(\tilde{y}|\tilde{x}) \quad (14)$$

Given a source sentence, the decoder predicts and outputs target sentences word by word. The decoding starts from an initial state. At each time step, the decoder selects top-N states (N is the beam size) and expands them until the end-of-sentence (EOS) symbol is produced. The final translation is generated by tracing back from the final state with the highest score.

We modified the decoder of GroundHog to perform translation under the log-linear framework. At each decoding state, the GroundHog only use the score estimated by RNN (Equation. 7) to select top-N candidates from the target vocabulary. In our decoder, we additionally calculate the word translation probabilities, the language model score and the current sentence length at each state. For each word in target vocabulary, we calculate a total score with Equation 6. Then we use the score to generate better candidate list since more features are used than the original NMT model. The weights of the log-linear models are tuned using the standard minimum-error-rate-training (MERT) (Och 2003) algorithm. To speed up the decoder, we use a priority queue to choose the best candidate to be extended (Hu et al. 2015) rather than extending all candidates at each state.

Experiments

We carried out experiments on Chinese-to-English translation. The training corpora are automatically crawled from the web, containing about 2.2 billion Chinese words and 2.3 billion English words. So far as we know, this is the largest amount of corpus that is used to train an NMT system.

System	DEV	TEST
GroundHog	36.72	30.61
+TM+WR	37.59	31.57
+LM	38.15	32.94
PBSMT	33.82	29.57

Table 1: BLEU scores on development and test sets. TM=translation model, WR=word reward, LM=language model, PBSMT=phrase based SMT.

We used NIST MT06 as the development set and tested our system on NIST MT08. The evaluation metric is case-insensitive BLEU-4 (Papineni et al., 2002). The feature weights of the translation system are tuned with the standard minimum-error-rate-training (MERT) (Och 2003) to maximize the systems' BLEU score on the development set.

We use the open-source NMT system, GroundHog (Bahdanau, Cho, and Bengio 2014), with default settings as our baseline system. We set beam size to 10 for decoding. As a comparison, we also report the performance of a phrase-based SMT (PBSMT) system, which is a re-implementation of the state-of-the-art phrase-based system, Moses (Koehn et al. 2007). Our SMT system is much more efficient both on training and decoding on our large bilingual corpus, and the translation quality is comparable with Moses. For the SMT system, we set the stack-limit to 200 and the translation-option-limit to 20.

Training

To train the *GroundHog* system, we limit the vocabulary to 30K most frequent words for both the source and target languages. Other words are replaced by a special symbol "UNK". The encoder consists of a forward RNN and a backward RNN, and each has 1000 hidden units. The decoder has 1000 hidden units. The word embeddings are 620-dimensional. A mini-batch stochastic gradient descent (SGD) together with Adadelta (Zeiler 2012) are used to train the networks. Each mini-batch of SGD contains 50 sentence pairs. Adadelta is used to adapt the learning rate of parameters ($\epsilon = 10^{-6}$ and $\rho = 0.95$). We ran both the training and decoding on a single machine with one GPU card (NVIDIA Tesla K10). The system is trained with about 1,570,000 updates for the RNN encoder.

For the *PBSMT* system, we obtained word alignment via the GIZA++ (Och and Ney 2004) and the "grow-diag-final" (Koehn, Och, and Marcu 2003) method. We trained a 5-gram language model (Stolcke 2002) with KN-discount on the target side of the bilingual corpus. The word translation table and the language model are then used as features being integrated with the *GroundHog* system.

Results

Table 1 lists the results on NIST test sets. We observed that the proposed method significantly improves the translation quality of the conventional NMT system. Moreover, our system outperforms the phrase-based SMT system on the same large training corpus.

Specifically, we can draw the following conclusions from Table 1:

1. By adding the word translation table and the word reward features, our method obtained significant improvements over the baseline (the results are shown in the row "*+TM+WR*"). There are three main reasons for the improvements. Firstly, the translation probabilities help the NMT system to perform better lexical translation. Secondly, the translation table is used to recover translations of unknown words. Thirdly, the word reward feature makes the decoder favors long translation. The average lengths of the outputs on the test set of our system and GroundHog are 23.5 and 21.4, respectively. This indicates that our method alleviates the inadequate translation problem. Further analyses and discussions will be described in the next Section.
2. Our method allows the NMT system to incorporate additional language models. We added a 5-gram language model trained on the target side of the bilingual corpus to the *GroundHog* system (the results are shown in the row "*+LM*"). It is observed that our method obtained further improvements on the test set, as the n-gram language model captures local target contextual information and improve the fluency.

Compared with the *GroundHog* system, our system (*GroundHog+TM+WR+LM*) achieves an absolute improvement of 2.33 points in BLEU score, which is statistically significant at p = 0.01 level (Riezler and Maxwell 2005).

Analysis and Discussion

In order to further study the performance of the proposed method, we compared the outputs of the systems.

Improving Lexical Translation

Taking the first sentence in Table 2 as an example, the GroundHog system omits the translation of "传输(chuanshu) *transmission*". In fact, the target words are in the vocabulary but not selected by the RNN model.

By integrating a translation table, our method produces the correct translation for the source words omited by the GroundHog. This can be attributed to the fact that the translation table consists of word pairs with translation probabilities estimated from the word-aligned training corpus, providing another way to measure the relationship between source and target words. In this example, the translation table contains the word pairs "*chuanshu, transmission*" with high probabilities.

To further improve the quality of lexical translation, we employ a conventional n-gram (n=5) language model to improve the local fluency. For example, there is another entry "*chuanshu, transfer*" for the Chinese word "*chuanshu*" in the translation table. The n-gram language model could help the decoder predict the correct translation, because $p_{lm}(transmission|series\quad of\quad high\quad speed)$ is greater than $p_{lm}(transfer|series\quad of\quad high\quad speed)$.

Source	没错，如同R400 笔记本电脑一样，接连一个可高速**传输** 的无线扩充槽。
PBSMT	Yes, like the R400 laptop, an expansion slot of a series of high speed wireless transmission.
GroundHog	Yes , like the R400 laptop , a series of high speed wireless expansion slot .
Our Method	Yes , like the R400 laptop , a series of high speed **transmission** of the wireless expansion slot .
Source	所有的人都被传染，热泪**横流** 。
PBSMT	All of the people have been infected, cross flow of tears.
GroundHog	All the people are infected , and tears UNK .
Our Method	All the people are infected, and tears **cross flow**.

Table 2: Translation examples. Chinese words in bold are correctly translated by our system.

System	OOV Percentage	
	DEV	TEST
PBSMT	1.6%	1.8%
GroundHog	4.4%	4.6%
Our Method	0.8%	0.9%

Table 3: Statistics of the percentages of the OOV words for the PBSMT, GroundHog and our method.

Translating the OOV Words

Table 3 shows the statistics of the OOV words for *PBSMT*, *GroundHog* and our systems on NIST06 and NIST08 test sets. It is observed that all the systems confront the OOV problem because the source words do not occur in the training corpus or the word pairs are not learned due to the word alignment error. However, the problem is much more serious for the *GroundHog* system since it limits the vocabulary size to reduce the model complexity.

The OOV words harm the translation quality. As shown in the second example in Table 2, the source word "ɬ(hengliu)" is not translated by *GroundHog*. After integrating the translation table within the log-linear framework, this word was correctly translated into "*cross flow*".

As demonstrated in Table 3, our method reduces about 82% of the OOV words for the NMT system. Moreover, the number of OOV words in our system is half of that in the PBSMT system. As we know, PBSMT system extracts word/phrase translations from word-aligned bilingual corpus. However, constrained by the inaccurate word alignment, not all words in the bilingual corpus are covered in the phrase table, causing OOV problems in the PBSMT system. Ideally, the RNN encoder-decoder is capable of translating all the words as long as they are encoded in the vocabulary. As the vocabulary used in the RNN encoder-decoder is limited for practical reasons, adding word translation table into RNN encoder-decoder combines the strength of both RNN and PBSMT, leading to a further reduction of OOV ratio.

Table 4 shows the effect of translating OOV words with the translation table. The row "*Our Method*" shares the same settings with Section *Experiment*. In these settings, the translation table is not only used to score word pairs, but also used to translate OOV words. The row "*-OOV*" means that the translation table is only used to score word pairs in the vocabulary of the RNN encoder-decoder, but not generate new translation candidates for the OOV words. It is observed that by translating OOV words, we obtained an absolute improvement of 0.37 points in BLEU scores on NIST08.

System	DEV	TEST
Our Method	38.15	32.94
-OOV	37.90	32.57

Table 4: Effect of translating OOV words. Our Method = GroundHog+TM+WR+LM, -OOV means the translation table is not used to recover OOV words.

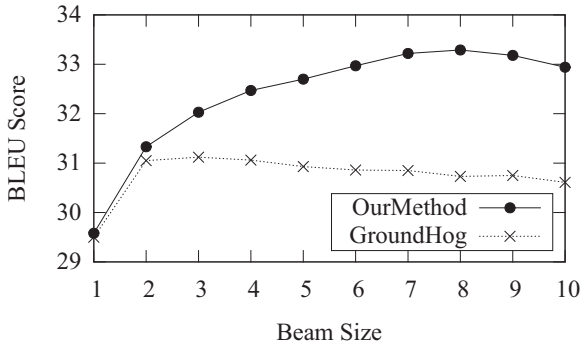

Figure 4: BLEU scores with respect to the beam sizes.

Improving Candidate List

Figure 4 shows the performance with respect to the beam size in the decoder. We vary beam size at test time while keeping the feature weights after MERT. We can see that the the BLEU score of GroundHog is not improved as the beam size increases after 3. On the other hand, the performance of the proposed method is improved with the increase of beam size. The reason is that, our method uses more features than GroundHog to generate candidate lists. With the beam size increasing, more better candidates are selected from the tar-

get vocabulary by the decoder.

Conclusion and Future Work

In this paper, we improve NMT by integrating additional SMT components (e.g. the translation table, the language model) under the log-linear framework, which makes the NMT approach be easily extended. The translation table is trained on word-aligned bilingual corpus via the standard phrase-based SMT method, and the language model is trained on monolingual target sentences. The proposed method alleviates major limitations of the current NMT architecture. The translation table recovers the omitted translations of source words and the OOV words, and the language model increases local fluency by making full use of monolingual corpus. Experiments on Chinese-to-English translation tasks show that our system achieves significant improvements over the baseline on large amount of the training corpus crawled from the web.

As a new approach, NMT still has more room for improvement. Current RNN encoder-decoder is actually a word-based translation system. In the future, we plan to improve NMT with phrase pairs, which are good at capturing local word reordering, idiom translation, etc.

Acknowledgements

This research is supported by the National Basic Research Program of China (973 program No. 2014CB340505). We would like to thank Xuan Liu and the anonymous reviewers for their insightful comments.

References

Auli, M.; Galley, M.; Quirk, C.; and Zweig, G. 2013. Joint language and translation modeling with recurrent neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 10441054.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. In *arXiv:1409.0473 [cs.CL]*.

Brown, P. F.; 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.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 17241734.

Devlin, J.; Zbib, R.; Huang, Z.; Lamar, T.; Schwartz, R.; and Makhoul, J. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 13701380.

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. In *arXiv:1503.03535 [cs.CL]*.

Hu, X.; Li, W.; Lan, X.; Wu, H.; and Wang, H. 2015. Optimized beam search with constrained softmax for nmt. In *MT Summit XV*.

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 demonstration session*.

Koehn, P.; Och, F. J.; and Marcu, D. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, 127–133.

Li, P.; Liu, Y.; and Sun, M. 2013. Recursive autoencoders for itg-based translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 567577.

Luong, M.-T.; Sutskever, I.; Le, Q. V.; Vinyals, O.; and Zaremba, W. 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)*, 11–19.

Och, F. J., and Ney, H. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 295–302.

Och, F. J., and Ney, H. 2004. The alignment template approach to statistical machine translation. 30:417–449.

Och, F. J. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 160–167.

Riezler, S., and Maxwell, J. T. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 5764.

Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *Signal Processing* IEEE Transactions on 45(11), 2673–2681.

Stolcke, A. 2002. Srilm – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken language Processing*, volume 2, 901–904.

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

Zeiler, M. D. 2012. Adadelta: An adaptive learning rate method. In *arXiv:1212.5701 [cs.LG]*.

Zhai, F.; Zhang, J.; Zhou, Y.; and Zong, C. 2013. Rnn-based derivation structure prediction for smt. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 779784.