Improving Neural Machine Translation through Phrase-based Forced Decoding

Jingyi Zhang[1,2], **Masao Utiyama**[1], **Eiichro Sumita**[1]
Graham Neubig[3,2], **Satoshi Nakamura**[2]
[1]National Institute of Information and Communications Technology, Japan
[2]Graduate School of Information Science, Nara Institute of Science and Technology, Japan
[3]Language Technologies Institute, Carnegie Mellon University, USA
jingyizhang/mutiyama/eiichiro.sumita@nict.go.jp
gneubig@cs.cmu.edu, s-nakamura@is.naist.jp

Abstract

Compared to traditional statistical machine translation (SMT), neural machine translation (NMT) often sacrifices adequacy for the sake of fluency. We propose a method to combine the advantages of traditional SMT and NMT by exploiting an existing phrase-based SMT model to compute the phrase-based decoding cost for an NMT output and then using this cost to rerank the n-best NMT outputs. The main challenge in implementing this approach is that NMT outputs may not be in the search space of the standard phrase-based decoding algorithm, because the search space of phrase-based SMT is limited by the phrase-based translation rule table. We propose a soft forced decoding algorithm, which can always successfully find a decoding path for any NMT output. We show that using the forced decoding cost to rerank the NMT outputs can successfully improve translation quality on four different language pairs.

1 Introduction

Neural machine translation (NMT), which uses a single large neural network to model the entire translation process, has recently been shown to outperform traditional statistical machine translation (SMT) such as phrase-based machine translation (PBMT) on several translation tasks (Koehn et al., 2003; Bahdanau et al., 2015; Sennrich et al., 2016a). Compared to traditional SMT, NMT generally produces more fluent translations, but often sacrifices adequacy, such as translating source words into completely unrelated target words, over-translation or under-translation (Koehn and Knowles, 2017).

There are a number of methods that combine the two paradigms to address their respective weaknesses. For example, it is possible to incorporate neural features into traditional SMT models to disambiguate hypotheses (Neubig et al., 2015; Stahlberg et al., 2016). However, the search space of traditional SMT is usually limited by translation rule tables, reducing the ability of these models to generate hypotheses on the same level of fluency as NMT, even after reranking. There are also methods that incorporate knowledge from traditional SMT into NMT, such as lexical translation probabilities (Arthur et al., 2016; He et al., 2016), phrase memory (Tang et al., 2016; Zhang et al., 2017), and n-gram posterior probabilities based on traditional SMT translation lattices (Stahlberg et al., 2017). These improve the adequacy of NMT outputs, but do not impose hard alignment constraints like traditional SMT systems and therefore cannot effectively solve all over-translation or under-translation problems.

In this paper, we propose a method that exploits an existing phrase-based translation model to compute the phrase-based decoding cost for a given NMT translation.[1] That is, we force a phrase-based translation system to take in the source sentence and generate an NMT translation. Then we use the cost of this phrase-based forced decoding to rerank the NMT outputs. The phrase-based decoding cost will heavily punish completely unrelated translations, over-translations, and under-translations, as they will not be able to be found in the translation phrase table.

One challenge in implementing this method is that the NMT output may not be in the search space of the phrase-based translation model, which is limited by the phrase-based translation

[1]In fact, our method can take in the output of *any* upstream system, but we experiment exclusively with using it to rerank NMT output.

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 152–162,
Taipei, Taiwan, November 27 – December 1, 2017

rule table. To solve this problem, we propose a soft forced decoding algorithm, which is based on the standard phrase-based decoding algorithm and integrates new types of translation rules (deleting a source word or inserting a target word). The proposed forced decoding algorithm can always successfully find a decoding path and compute a phrase-based decoding cost for any NMT output. Another challenge is that we need a diverse NMT n-best list for reranking. Because beam search for NMT often lacks diversity in the beam – candidates only have slight differences, with most of the words overlapping – we use a random sampling method to obtain a more diverse n-best list.

We test the proposed method on English-to-Chinese, English-to-Japanese, English-to-German and English-to-French translation tasks, obtaining large improvements over a strong NMT baseline that already incorporates discrete lexicon features.

2 Attentional NMT

Our baseline NMT model is similar to the attentional model of Bahdanau et al. (2015), which includes an encoder, a decoder and an attention (alignment) model. Given a source sentence $F = \{f_1, ..., f_J\}$, the encoder learns an annotation $h_j = \left[\vec{h}_j; \overleftarrow{h}_j\right]$ for f_j using a bi-directional recurrent neural network.

The decoder generates the target translation from left to right. The probability of generating next word e_i is,[2]

$$P_{NMT}\left(e_i|e_1^{i-1}, F\right) = softmax\left(g\left(e_{i-1}, t_i, s_i\right)\right) \quad (1)$$

where t_i is a decoding state for time step i, computed by,

$$t_i = f\left(t_{i-1}, e_{i-1}, s_i\right) \quad (2)$$

s_i is a source representation for time i, calculated as,

$$s_i = \sum_{j=1}^{J} \alpha_{i,j} \cdot h_j \quad (3)$$

where $\alpha_{i,j}$ scores how well the inputs around position j and the output at position i match, computed as,

$$\alpha_{i,j} = \frac{\exp\left(a\left(t_{i-1}, h_j\right)\right)}{\sum_{k=1}^{J} \exp\left(a\left(t_{i-1}, h_k\right)\right)} \quad (4)$$

[2] g, f and a in Equation 1, 2 and 4 are nonlinear, potentially multi-layered, functions.

As we can see, NMT only learns an attention (alignment) distribution for each target word over all source words and does not provides exact mutually-exclusive word or phrase level alignments. As a result, it is known that attentional NMT systems make mistakes in over- or under-translation (Cohn et al., 2016; Mi et al., 2016).

3 Phrase-based Forced Decoding for NMT

3.1 Phrase-based SMT

In phrase-based SMT (Koehn et al., 2003), a phrase-based translation rule r includes a source phrase, a target phrase and a translation score $S(r)$. Phrase-based translation rules can be extracted from the word-aligned training set and then used to translate new sentences. Word alignments for the training set can be obtained by IBM models (Brown et al., 1993).

Phrase-based decoding uses a list of translation rules to translate source phrases in the input sentence and generate target phrases from left to right. A basic concept in phrase-based decoding is hypotheses. As shown in Figure 1, the hypothesis H_1 consists of two rules r_1 and r_2. The score of a hypothesis $S(H)$ can be calculated as the product of the scores of all applied rules.[3] An existing hypothesis can be expanded into a new hypothesis by applying a new rule. As shown in Figure 1, H_1 can be expanded into H_2, H_3 and H_4. H_2 cannot be further expanded, because it covers all source words, while H_3 and H_4 can (and must) be further expanded. The decoder starts with an initial empty hypothesis H_0 and selects the hypothesis with the highest score from all completed hypotheses.

During decoding, hypotheses are stored in stacks. For a source sentence with J words, the decoder builds J stacks. The hypotheses that cover j source words are stored in stack s_j. The decoder expands hypotheses in $s_1, s_2, ..., s_J$ in turn as shown in Algorithm 1. Here, EXPAND(H) is expanding H to get new hypotheses and putting the new hypotheses into corresponding stacks. For each stack, a beam of the best n hypotheses is kept to speed up the decoding process.

[3] In actual phrase-based decoding it is common to integrate reordering probabilities in the forced decoding score defined in Equation 9. However, because NMT generally produces more properly ordered sentences than traditional SMT, in this work we do not consider reordering probabilities in our forced decoding algorithm.

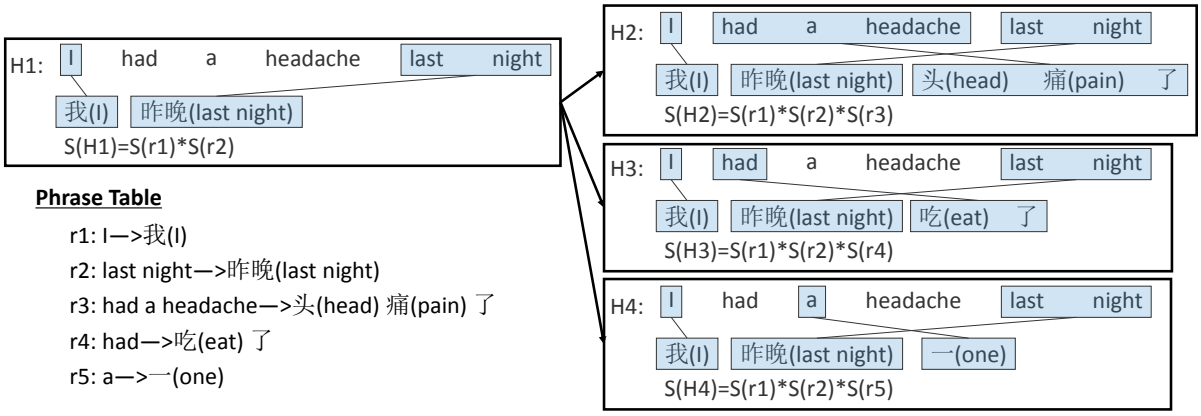

Figure 1: An example of phrase-based decoding.

Algorithm 1 Standard phrase-based decoding.

Require: Source sentence F with length J
Ensure: Translation E and decoding path D
  initialize H_0 and $s_1, s_2, ..., s_J$
  EXPAND(H_0)
  for $j = 1$ to $J - 1$ **do**
    for each hypothesis H_{jk} in s_j **do**
      EXPAND(H_{jk})
  select best hypothesis in s_J

3.2 Forced Decoding for NMT

As stated in the introduction, our goal is not to generate new hypotheses with phrase-based SMT, but instead use the phrase-based model to calculate scores for NMT output. In order to do so, we can perform *forced decoding*, which is very similar to the algorithm in the previous section but discards all partial hypotheses that do not match the NMT output. However, the NMT output is not limited by the phrase-based rule table, so there may be no decoding path that completely matches the NMT output when using only the phrase-based rules.

To remedy this problem, inspired by previous work in forced decoding for training phrase-based SMT systems (Wuebker et al., 2010, 2012) we propose a soft forced decoding algorithm that can always successfully find a decoding path for a source sentence F and an NMT translation E.

First, we introduce two new types of rules R_1 and R_2.

R_1 A source word f can be translated into a special word `null`. This corresponds to deleting f during translation. The score of deleting f is calculated as,

$$s(f \rightarrow \texttt{null}) = \frac{\text{unalign}(f)}{|\mathcal{T}|} \quad (5)$$

where unalign (f) is how many times f is unaligned in the word-aligned training set $\mathcal{T}$ and $|\mathcal{T}|$ is the number of sentence pairs in $\mathcal{T}$.

R_2 A target word e can be translated from a special word `null`, which corresponds to inserting e during translation. The score of inserting e is calculated as,

$$s(\texttt{null} \rightarrow e) = \frac{\text{unalign}(e)}{|\mathcal{T}|} \quad (6)$$

where unalign (e) is how many times e is unaligned in $\mathcal{T}$.

One motivation for Equations 5 and 6 is that function words usually have high frequencies, but do not have as clear a correspondence with a word in the other language as content words. As a result, in the training set function words are more often unaligned than content words. As an example, Table 1 and Table 2 show how many times different words occur and how many times they are unaligned in the word-aligned training set of English-to-Chinese and English-to-French tasks in our experiments. As we can see, generally there are less unaligned words in the English-to-French task, however, function words are more likely to be unaligned in both tasks. Based on Equation 5 and Equation 6, the scores of deleting or inserting "of" and "a" will be higher.

In our forced decoding, we choose to model the score of each translation rule that exists in the phrase table as the product of direct and inverse phrase translation probabilities. To make sure that

Words	of	a	practice	water
Occur	1.3M	1.0M	2.2K	29K
Unaligned	0.51M	0.41M	0.25K	3.5K

Table 1: The number of times that words occur in the English-to-Chinese training corpus and the number of times that they are unaligned.

Words	of	a	practice	water
Occur	1.7M	0.83M	8.8K	7.4K
Unaligned	0.16M	0.12M	0.38K	0.19K

Table 2: The number of times that words occur in the English-to-French training corpus and the number of times that they are unaligned.

the scale of the scores for R_1 and R_2 match the other phrase (which are the product of two probabilities), we use the square of the score in Equation 5/6 as the rule score for R_1/R_2.

Algorithm 2 shows the forced decoding algorithm that integrates the new rules. Because the translation E is given for the forced decoding algorithm, the proposed forced decoding algorithm keeps I stacks, where I is the length of E. In other words, the stack size is corresponding to the target word size during forced decoding while the stack size is corresponding to the source word size during standard phrase-based decoding. The stack s'_i in Algorithm 2 contains all hypotheses in which the first i target words have been generated. We expand hypotheses in $s'_1, s'_2, ..., s'_I$ in turn. When expanding a hypothesis H_{old} in s'_i, besides expanding it using the original rule table EXPAND(H_{old}),[4] we also expand H_{old} by inserting the next target word e_{i+1} at the end of H_{old} to get an additional hypothesis H_{new} and put H_{new} into s'_{i+1}. For a final hypothesis in stack s'_I, it may not cover all source words. We update its score by translating uncovered words into `null`.

Because different decoding paths can generate the same final translation, there can be different decoding paths that fit the NMT translation E. We use the score of the single decoding path with the highest decoding score as the forced decoding score for E.

[4]The new introduced word inserting/deleting rules are not used when performing EXPAND(H_{old}).

Algorithm 2 Forced phrase-based decoding.

Require: Source sentence F with length J and translation E with length I
Ensure: Decoding path D
  initialize H_0 and $s'_1, s'_2, ..., s'_I$
  EXPAND(H_0)
  expand H_0 with rule `null`$\rightarrow e_1$
  for $i = 1$ to $I - 1$ **do**
    for each hypothesis H_{ik} in s'_i **do**
      EXPAND(H_{ik})
      expand H_{ik} with rule `null`$\rightarrow e_{i+1}$
  for each hypothesis H_{Ik} in s'_I **do**
    update $S(H_{Ik})$ for uncovered source words
  select best hypothesis in s'_I

4 Reranking NMT Outputs with Phrase-based Decoding Score

We rerank the n-best NMT outputs using the phrase-based forced decoding score according to Equation 7.

$$\log P(E|F) = w_1 \cdot \log P_n(E|F) + w_2 \cdot \log S_d(E|F) \quad (7)$$

where $P_n(E|F)$ is the original NMT translation probability as calculated by Equation 1;

$$P_n(E|F) = \prod_{i=1}^{I} P_{NMT}\left(e_i|e_1^{i-1}, F\right) \quad (8)$$

$S_d(E|F)$ is the forced decoding score, which is the score of the decoding path $\hat{D}$ with the highest decoding score as described above;

$$S_d(E|F) = \prod\nolimits_{r \in \hat{D}} S(r) \quad (9)$$

w_1 and w_2 are weights that can be tuned on the n-best list of the development set.

The easiest way to get an n-best list for NMT is by using the n-best translations from beam search, which is the standard decoding algorithm for NMT. While beam search is likely to find the highest-scoring hypothesis, it often lacks diversity in the beam: candidates only have slight differences, with most of the words overlapping. In order to obtain a more diverse list of hypotheses for reranking, we additionally augment the 1-best hypothesis discovered by beam search with translations sampled from the NMT conditional probability distribution.

The standard method for sampling hypotheses in NMT is ancestral sampling, where we randomly select a word from the vocabulary according to

$P_{NMT}\left(e_i|e_1^{i-1},F\right)$ (Shen et al., 2016). This will make a diverse list of hypotheses, but may reduce the probability of selecting a highly scoring hypothesis, and the whole n-best list may not contain any candidate with better translation quality than the standard beam search output.

Instead, we take an alternative approach that proved empirically better in our experiments: at each time step i, we use sampling to randomly select the next word from e' and e'' according to Equation 10. Here, e' and e'' are the two target words with the highest probability according to Equation 1.

$$P_{rdm}\left(e'\right) = \frac{P_{NMT}\left(e'|e_1^{i-1},F\right)}{P_{NMT}\left(e'|e_1^{i-1},F\right)+P_{NMT}\left(e''|e_1^{i-1},F\right)}$$
$$P_{rdm}\left(e''\right) = \frac{P_{NMT}\left(e''|e_1^{i-1},F\right)}{P_{NMT}\left(e'|e_1^{i-1},F\right)+P_{NMT}\left(e''|e_1^{i-1},F\right)} \quad (10)$$

The sampling process ends when $\langle/s\rangle$ is selected as the next word.

We repeat the decoding process $1,000$ times to sample $1,000$ outputs for each source sentence. We additionally add the 1-best output of standard beam search, making the size of the list used for reranking to be $1,001$.

5 Experiments

5.1 Settings

We evaluated the proposed approach for English-to-Chinese (en-zh), English-to-Japanese (en-ja), English-to-German (en-de) and English-to-French (en-fr) translation tasks. For the en-zh and en-ja tasks, we used datasets provided for the patent machine translation task at NTCIR-9 (Goto et al., 2011).[5] For the en-de and en-fr tasks, we used version 7 of the Europarl corpus as training data, WMT 2014 test sets as our development sets and WMT 2015 test sets as our test sets. The detailed statistics for training, development and test sets are given in Table 3. The word segmentation was done by BaseSeg (Zhao et al., 2006) for Chinese and Mecab[6] for Japanese.

We built attentional NMT systems with Lamtram[7]. Word embedding size and hidden layer size are both 512. We used Byte-pair encoding (BPE) (Sennrich et al., 2016b) and set the vocabulary size to be 50K. We used the Adam algorithm for optimization.

To obtain a phrase-based translation rule table for our forced decoding algorithm, we used GIZA++ (Och and Ney, 2003) and *grow-diag-final-and* heuristic to obtain symmetric word alignments for the training set. Then we extracted the rule table using Moses (Koehn et al., 2007).

			SOURCE	TARGET
en-de	TRAIN	#Sents	1.90M	
		#Words	52.2M	49.7M
		#Vocab	113K	376K
	DEV	#Sents	3,003	
		#Words	67.6K	63.0K
	TEST	#Sents	2,169	
		#Words	46.8K	44.0K
en-fr	TRAIN	#Sents	1.99M	
		#Words	54.4M	60.4M
		#Vocab	114K	137K
	DEV	#Sents	3,003	
		#Words	71.1K	81.1K
	TEST	#Sents	1.5K	
		#Words	27.1K	29.8K
en-zh	TRAIN	#Sents	954K	
		#Words	40.4M	37.2M
		#Vocab	504K	288K
	DEV	#Sents	2K	
		#Words	77.5K	75.4K
	TEST	#Sents	2K	
		#Words	58.1K	55.5K
en-ja	TRAIN	#Sents	3.14M	
		#Words	104M	118M
		#Vocab	273K	150K
	DEV	#Sents	2K	
		#Words	66.5K	74.6K
	TEST	#Sents	2K	
		#Words	70.6K	78.5K

Table 3: Data sets.

5.2 Results and Analysis

Table 4 shows results of the phrase-based SMT system[8], the baseline NMT system, the lexicon integration method (Arthur et al., 2016) and the proposed reranking method. We tested three features for reranking: the NMT score P_n, the forced decoding score S_d and a word penalty (WP) feature, which is the length of the translation. The best NMT system and the systems that have no significant difference from the best NMT system at the $p < 0.05$ level using bootstrap resampling (Koehn, 2004) are shown in bold font.

As we can see, integrating lexical translation probabilities improved the baseline NMT system

[5]Note that NTCIR-9 only contained a Chinese-to-English translation task, we used English as the source language in our experiments. In NTCIR-9, the development and test sets were both provided for the zh-en task while only the test set was provided for the en-ja task. We used the sentences from the NTCIR-8 en-ja and ja-en test sets as the development set in our experiments.

[6]http://sourceforge.net/projects/mecab/files/

[7]https://github.com/neubig/lamtram

[8]We used the default Moses settings for phrase-based SMT.

	en-zh		en-ja		en-de		en-fr	
	dev	test	dev	test	dev	test	dev	test
PBMT	30.73	27.72	35.67	33.46	12.37	13.95	25.96	27.50
NMT	34.60	32.71	41.67	39.00	12.52	14.05	23.63	23.99
NMT+lex	36.06	34.80	44.47	41.09	13.36	15.60	24.00	24.91
NMT+lex+rerank(P_n)	34.38	33.23	38.92	34.18	12.34	13.59	23.13	23.61
NMT+lex+rerank(S_d)	36.17	34.09	42.91	40.16	13.08	15.29	24.28	25.71
NMT+lex+rerank(P_n+S_d)	37.94	**35.59**	45.34	41.75	**14.56**	**16.61**	**25.96**	**27.12**
NMT+lex+rerank(P_n+WP)	37.44	34.93	45.81	41.90	13.75	15.46	24.47	25.09
NMT+lex+rerank(S_d+WP)	36.44	33.73	43.52	40.49	13.39	15.71	24.74	26.25
NMT+lex+rerank(P_n+S_d+WP)	**38.69**	**35.75**	**46.92**	**43.17**	**14.61**	**16.65**	**25.98**	**27.15**

Table 4: Translation results (BLEU). NMT+lex: (Arthur et al., 2016); NMT+lex+rerank: we rerank the n-best outputs of NMT+lex using different features (P_n, S_d and WP).

	en-zh		en-ja		en-de		en-fr	
	METEOR	chrF	METEOR	chrF	METEOR	chrF	METEOR	chrF
PBMT	34.70	37.87	35.22	39.45	26.66	50.02	32.33	56.36
NMT	34.51	39.91	35.07	42.02	24.91	44.50	29.58	49.99
NMT+lex	35.56	42.22	36.48	44.34	25.49	45.67	30.10	50.89
NMT+lex+rerank(P_n)	34.56	40.80	32.63	38.57	23.57	40.35	29.15	48.64
NMT+lex+rerank(S_d)	36.02	42.65	36.87	44.85	**26.48**	**48.73**	**31.56**	**54.42**
NMT+lex+rerank(P_n+S_d)	36.40	43.73	37.22	45.69	**26.26**	47.27	**31.62**	53.99
NMT+lex+rerank(P_n+WP)	36.04	42.86	36.90	44.93	25.03	44.05	30.21	50.78
NMT+lex+rerank(S_d+WP)	36.34	42.78	37.05	45.03	26.16	47.82	31.32	53.75
NMT+lex+rerank(P_n+S_d+WP)	**36.88**	**44.09**	**37.94**	**46.66**	**26.20**	47.12	**31.61**	53.98

Table 5: METEOR and chrF scores on the test sets for different system outputs in Table 4.

	en-zh		en-ja		en-de		en-fr	
	dev	test	dev	test	dev	test	dev	test
PBMT	1.008	1.018	1.005	0.998	1.077	1.069	0.986	1.004
NMT	0.953	0.954	0.960	0.961	1.059	1.038	0.985	0.977
NMT+lex	0.936	0.966	0.955	0.963	1.054	1.019	1.030	0.977
NMT+lex+rerank(P_n)	0.875	0.898	0.814	0.775	0.874	0.854	0.904	0.900
NMT+lex+rerank(S_d)	0.973	0.989	0.985	0.981	1.062	1.060	1.030	1.031
NMT+lex+rerank(P_n+S_d)	0.949	0.965	0.945	0.936	1.000	0.992	0.999	0.992
NMT+lex+rerank(P_n+WP)	0.996	1.019	0.999	0.983	1.000	0.975	0.998	1.001
NMT+lex+rerank(S_d+WP)	1.000	1.024	1.001	1.001	1.011	1.007	0.999	0.989
NMT+lex+rerank(P_n+S_d+WP)	0.990	1.014	1.000	0.986	1.000	0.989	1.000	0.992

Table 6: Ratio of translation length to reference length for different system outputs in Table 4.

and reranking with the three features all together achieved further improvements for all four language pairs. Even on English-to-Chinese and English-to-Japanese tasks, where the NMT system outperformed the phrase-based SMT system by 7-8 BLEU scores, using the forced decoding score for reranking NMT outputs can still achieve significant improvements. With or without the word penalty feature, using both P_n and S_d for reranking gave better results than only using P_n or S_d alone. We also show METEOR and chrF scores on the test sets in Table 5. Our reranking method improved both METEOR and chrF significantly.

The Word Penalty Feature The word penalty feature generally improved the reranking results, especially when only the NMT score P_n was used for reranking. As we can see, using only P_n for reranking decreased the translation quality compared to the standard beam search result of NMT. Because the search spaces of beam search and random sampling are quite different, the best beam search output does not necessarily have the highest NMT score compared to random sampling outputs. Therefore, even the P_n reranking results do have higher NMT scores, but have lower BLEU scores according to Table 4. To explain why this happened, we show the ratio of translation length to reference length in Table 6. As we can see, the P_n reranking outputs are much shorter. This is because NMT generally prefers shorter translations, since Equation 8 multiplies all target word probabilities together. So the word penalty feature can improve the P_n reranking results considerably, by preferring longer sentences. Because the forced decoding score S_d as shown in Equation 9 does not obviously prefer shorter or longer sentences, when S_d was used for reranking, the word penalty

Source	for hypophysectomized (hypop hy sec to mized) rats , the drinking water additionally contains 5 % glucose .
Reference	对于(for) 去(remove) 垂体(hypophysis) 大(big) 鼠(rat) ， 饮用水(drinking water) 中(in) 另外(also) 含有(contain) 5 % 葡萄糖(glucose) 。
PBMT	用于(for) 大(big) 鼠(rat) 垂体(hypophysis) HySecto，(Hy Sec to ，) 饮用水(drinking water) 另外(also) 含有(contain) 5 % 葡萄糖(glucose) 。
NMT	对于(for) 过(pass) 盲肠(cecum) 的(of) 大(big) 鼠(rat) ， 饮用水(drinking water) 另外(also) 含有(contain) 5 % 葡萄糖(glucose) 。
NMT+lex NMT+lex+P_n NMT+lex+P_n+WP	对于(for) 低(low) 酪(cheese) 蛋白(protein) 切除(remove) 的(of) 大(big) 鼠(rat) ， 饮用水(drinking water) 另外(also) 含有(contain) 5 % 葡萄糖(glucose) 。
NMT+lex+S_d NMT+lex+S_d+WP	对于(for) 垂体(hypophysis) 在(is) 切除(remove) 大(big) 鼠(rat) 中(in) ， 饮用水(drinking water) 另外(also) 含有(contain) 5 % 葡萄糖(glucose) 。
NMT+lex+P_n+S_d NMT+lex+P_n+S_d+WP	对于(for) 垂体(hypophysis) 在(is) 切除(remove) 的(of) 大(big) 鼠(rat) 中(in) ， 饮用水(drinking water) 另外(also) 含有(contain) 5 % 葡萄糖(glucose) 。

Table 7: An example of improving inaccurate rare word translation by using S_d for reranking.

feature became less helpful. When both P_n and S_d were used for reranking, the word penalty feature only achieved further significant improvement on the English-to-Japanese task.

T_1 (NMT+lex):	
for →对于(for)	-3.04
r_a: hy →低(low)	-12.19
r_b: null→酪(cheese)	-21.99
r_c: null→蛋白(protein)	-13.83
to mized →切除(remove)	-6.22
null→的(of)	-1.53
rats →大(big) 鼠(rat)	-1.52
, the drinking water →， 饮用水(drinking water)	-1.38
additionally contains →另外(also) 含有(contain)	-3.68
5 % →5 %	-0.51
glucose . →葡萄糖(glucose) 。	-0.60
r_d: hypop→null	-25.33
sec→null	-20.66
T_2 (NMT+lex+P_n+S_d):	
for →对于(for)	-3.04
hypop hy →垂体(hypophysis)	-5.09
the →在(is)	-5.32
to mized →切除(remove)	-6.22
null→的(of)	-1.53
rats →大(big) 鼠(rat)	-1.52
, →中(in) ,	-4.11
drinking water →饮用水(drinking water)	-1.03
additionally contains →另外(also) 含有(contain)	-3.68
5 % →5 %	-0.51
glucose . →葡萄糖(glucose) 。	-0.60
sec→null	-20.66

Table 9: Forced decoding paths for T_1 and T_2: used rules and log scores. The translation rules with shade are used only for T_1 or T_2.

Table 7 gives translation examples of our reranking method from the English-to-Chinese task. The source English word "hypophysectomized" is an unknown word which does not occur in the training set. By employing BPE, this word is split into "hypop", "hy", "sec", "to" and "mized". The correct translation for "hypophysectomized" is "去(remove) 垂体(hypophysis)" as shown in the reference sentence. The original attentional NMT translated it into incorrect translation "过(pass) 盲肠(cecum)". After integrating lexicons, the NMT system translated it into "低(low) 酪(cheese) 蛋白(protein) 切除(remove)". The last word "切除(remove)" is correct, but the rest of the translation is still wrong. Only by using the forced decoding score S_d for reranking, we get the more accurate translation "垂体(hypophysis) 在(is) 切除(remove)".

To further demonstrate how the reranking method works, Table 9 shows translation rules and their log-scores contained in the forced decoding paths found for T_1, the NMT translation without reranking and T_2, the NMT translation using both P_n and S_d for reranking. As we can see, the four rules r_a, r_b, r_c and r_d used for T_1 have low scores. r_a is an unlikely translation. In r_b, r_c and r_d, "酪(cheese)", "蛋白(protein)" and "hypop" are content words, which are unlikely to be deleted or inserted during translation. Table 9 also shows that the translation of function words is very flexible. The score of inserting a function word "的(of)" is very high. The translation rule "the →在(is)" used for T_2 is incorrect, but its score is relatively high, because function words are often

Source	such changes in reaction conditions include , but are not limited to , an increase in temperature or change in ph .
Reference	所(such) 述(said) 反 应(reaction) 条 件(condition) 的(of) 改 变(change) 包 括(include) 但(but) 不(not) 限 于(limit) 温度(temperature) 的(of) 增加(increase) 或(or) pH 值(value) 的(of) 改变(change) 。
PBMT	中(in) 的(of) 这 种(such) 变 化(change) 的(of) 反 应(reaction) 条 件(condition) 包 括(include) , 但(but) 不(not) 限 于(limit) , 增加(increase) 的(of) 温度(temperature) 或(or) pH 变化(change) 。
NMT	这种(such) 反应(reaction) 条件(condition) 的(of) 变化(change) 包括(include) 但(but) 不(not) 限于(limit) pH 或(or) pH 的(of) 变化(change) 。
NMT+lex NMT+lex+P_n	这种(such) 反应(reaction) 条件(condition) 的(of) 变化(change) 包括(include) , 但(but) 不(not) 限于(limit) , pH 的(of) 升高(increase) 或(or) pH 变化(change) 。
NMT+lex+S_d	这种(such) 反应(reaction) 条件(condition) 的(of) 变化(change) 包括(include) 但(but) 不(not) 限于(limit) , 温度(temperature) 的(of) 升高(increase) 或(or) 改变(change) pH 值(value) 。
NMT+lex+P_n+S_d	这种(such) 反应(reaction) 条件(condition) 的(of) 变化(change) 包括(include) , 但(but) 不(not) 限于(limit) , 温度(temperature) 的(of) 升高(increase) 或(or) 改变(change) pH 值(value) 。
NMT+lex+P_n+WP	这种(such) 反应(reaction) 条件(condition) 的(of) 变化(change) 包括(include) , 但(but) 不(not) 限于(limit) , pH 的(of) 升高(increase) 或(or) 改变(change) pH 值(value) 。
NMT+lex+S_d+WP NMT+lex+P_n+S_d+WP	这种(such) 反应(reaction) 条件(condition) 的(of) 变化(change) 包括(include) , 但(but) 不(not) 限于(limit) , 温度(temperature) 的(of) 升高(increase) 或(or) 改变(change) pH 值(value) 。

Table 8: An example of improving under-translation and over-translation by using S_d for reranking.

incorrectly aligned in the training set. The reason why function words are more likely to be incorrectly aligned to each other is that they usually have high frequencies and do not have clear correspondences between different languages.

In T_1, "hypophysectomized (hypop hy sec to mized)" is incorrectly translated into "低(low) 酪(cheese) 蛋白(protein) 切除(remove)". However, from Table 9, we can see that the forced decoding algorithm learns it as unlikely translation (hy→低(low)), over-translation (`null`→酪(cheese), `null`→蛋白(protein)) and under-translation (hypop→`null`, sec→`null`), because there is no translation rule between "hypop" "sec" and "酪(cheese)" "蛋白(protein)". Because content words are unlikely to be deleted or inserted during translation, they have low forced decoding scores. So using the forced decoding score for reranking NMT outputs can naturally improve over-translation or under-translation as shown in Table 8. As we can see, without using S_d for reranking, NMT under-translated "temperature" and over-translated "ph" twice, which will be assigned low scores by forced decoding. By using S_d for reranking, the correct translation was selected.

We did human evaluation on 100 sentences randomly selected from the English-to-Chinese test set to test the effectiveness of our forced decoding method. We compared the outputs of two systems:

- NMT+lex+rerank(P_n+WP)
- NMT+lex+rerank(P_n+S_d+WP)

For each source sentence, we compared the two system outputs. Table 10 shows the numbers of sentences that our forced decoding feature helped to reduce completely unrelated translation, over-translation and under-translation. The last line of Table 10 means that for 73 source sentences, our forced decoding feature neither reduced nor caused more unrelated/over/under translation. That is our forced decoding feature never caused more unrelated/over/under translation for the sampled 100 sentences, which shows that our method is very robust for improving unrelated/over/under translation.

Reduce	both under- and over- translation	2
	under-translation	11
	over-translation	10
	unrelated translation	4
No difference		73

Table 10: Human evaluation results.

Reranking PBMT Outputs with NMT We also did experiments that use the NMT score as an additional feature to rerank PBMT outputs (unique $1,000$-best list). The results are shown

in Table 11. We also copy results of baseline PBMT and NMT from Table 4 for direct comparison. As we can see, using NMT to rerank PBMT outputs achieved improvements over the baseline PBMT system. However, when the baseline NMT system is significantly better than the baseline PBMT system (en-zh, en-ja), even using NMT to rerank PBMT outputs still achieved lower translation quality compared to the baseline NMT system.

		en-zh	en-ja	en-de	en-fr
PBMT+rerank		32.77	37.68	**14.23**	**28.86**
PBMT	dev	30.73	35.67	12.37	25.96
NMT		**34.60**	**41.67**	12.52	23.63
PBMT+rerank		30.04	35.14	**15.89**	**29.77**
PBMT	test	27.72	33.46	13.95	27.50
NMT		**32.71**	**39.00**	14.05	23.99

Table 11: Results of using NMT for reranking PBMT outputs.

6 Related Work

Wuebker et al. (2010, 2012) applied forced decoding on the training set to improve the training process of phrase-based SMT and prune the phrase-based rule table. They also used word insertions and deletions for forced decoding, but they used a high penalty for all insertions and deletions. In contrast, our soft forced decoding algorithm for NMT outputs uses a small penalty for function words and a high penalty for content words, because function words are usually translated very flexibly and more likely to be inserted or deleted compared to content words. For example, the under-translation of a content word can hurt the adequacy of the translation heavily. But function words may naturally disappear during translation (e.g. the English word "the" disappears in Chinese). By assigning a high penalty to words that should not be deleted or inserted during translation, our soft forced decoding method aims to improve the adequacy of NMT, which is very different from previous forced decoding methods that are used to improve general SMT training (Yu et al., 2013; Xiao et al., 2016).

A major difference of traditional SMT and NMT is that the alignment model in traditional SMT provides exact word or phrase level alignments between the source and target sentences while the attention model in NMT only computes an alignment probability distribution for each target word over all source words, which is the main reason why NMT is more likely to produce completely unrelated translations, over-translation or under-translation compared to traditional SMT. To relieve NMT of these problems, there are methods that modify the NMT neural network structure (Tu et al., 2016; Meng et al., 2016; Alkhouli et al., 2016) while we rerank NMT outputs by exploiting knowledge from traditional SMT.

There are also existing methods that rerank NMT outputs by using target-bidirectional NMT models (Liu et al., 2016; Sennrich et al., 2016a). Their reranking method aims to overcome the issue of unbalanced accuracy in NMT outputs while our reranking method aims to solve the inadequacy problem of NMT.

7 Conclusion

In this paper, we propose to exploit an existing phrase-based SMT model to compute the phrase-based decoding cost for NMT outputs and then use the phrase-based decoding cost to rerank the n-best NMT outputs, so we can combine the advantages of both PBMT and NMT. Because an NMT output may not be in the search space of standard phrase-based SMT, we propose a forced decoding algorithm, which can always successfully find a decoding path for any NMT output by deleting source words and inserting target words. Results show that using the forced decoding cost to rerank NMT outputs improved translation accuracy on four different language pairs.

References

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 54–65.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K Tsou. 2011. Overview of the patent machine translation task at the NTCIR-9 workshop. In *Proc. NTCIR-9*, pages 559–578.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with SMT features. In *Thirtieth AAAI conference on artificial intelligence*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416.

Fandong Meng, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Interactive attention for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2174–2185.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960.

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692.

Felix Stahlberg, Adrià de Gispert, Eva Hasler, and Bill Byrne. 2017. Neural machine translation by minimising the Bayes-risk with respect to syntactic translation lattices. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 362–368.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. 2016. Syntactically guided neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 299–305.

Yaohua Tang, Fandong Meng, Zhengdong Lu, Hang Li, and Philip LH Yu. 2016. Neural machine translation with external phrase memory. *arXiv preprint arXiv:1606.01792*.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85.

Joern Wuebker, Mei-Yuh Hwang, and Chris Quirk. 2012. Leave-one-out phrase model training for large-scale deployment. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 460–467.

Joern Wuebker, Arne Mauser, and Hermann Ney. 2010. Training phrase translation models with leaving-one-out. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 475–484.

Tong Xiao, Derek F Wong, and Jingbo Zhu. 2016. A loss-augmented approach to training syntactic machine translation systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2069–2083.

Heng Yu, Liang Huang, Haitao Mi, and Kai Zhao. 2013. Max-violation perceptron and forced decoding for scalable MT training. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1112–1123.

Jiacheng Zhang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. 2017. Prior knowledge integration for neural machine translation using posterior regularization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1514–1523.

Hai Zhao, Chang-Ning Huang, Mu Li, et al. 2006. An improved chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165.