Guiding Neural Machine Translation with Retrieved Translation Pieces

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

One of the difficulties of neural machine translation (NMT) is the recall and appropriate translation of low-frequency words or phrases. In this paper, we propose a simple, fast, and effective method for recalling previously seen translation examples and incorporating them into the NMT decoding process. Specifically, for an input sentence, we use a search engine to retrieve sentence pairs whose source sides are similar with the input sentence, and then collect n-grams that are both in the retrieved target sentences and aligned with words that match in the source sentences, which we call "translation pieces". We compute pseudo-probabilities for each retrieved sentence based on similarities between the input sentence and the retrieved source sentences, and use these to weight the retrieved translation pieces. Finally, an existing NMT model is used to translate the input sentence, with an additional bonus given to outputs that contain the collected translation pieces. We show our method improves NMT translation results up to 6 BLEU points on three narrow domain translation tasks where repetitiveness of the target sentences is particularly salient. It also causes little increase in the translation time, and compares favorably to another alternative retrieval-based method with respect to accuracy, speed, and simplicity of implementation.

1 Introduction

Neural machine translation (NMT) (Bahdanau et al., 2014; Sennrich et al., 2016a; Wang et al., 2017b) is now the state-of-the-art in machine translation, due to its ability to be trained end-to-end on large parallel corpora and capture complex parameterized functions that generalize across a variety of syntactic and semantic phenomena. However, it has also been noted that compared to alternatives such as phrase-based translation (Koehn et al., 2003), NMT has trouble with low-frequency words or phrases (Arthur et al., 2016; Kaiser et al., 2017), and also generalizing across domains (Koehn and Knowles, 2017). A number of methods have been proposed to ameliorate these problems, including methods that incorporate symbolic knowledge such as discrete translation lexicons (Arthur et al., 2016; He et al., 2016; Chatterjee et al., 2017) and phrase tables (Zhang et al., 2017; Tang et al., 2016; Dahlmann et al., 2017), adjust model structures to be more conducive to generalization (Nguyen and Chiang, 2017), or incorporate additional information about domain (Wang et al., 2017a) or topic (Zhang et al., 2016) in translation models.

In particular, one paradigm of interest is recent work that augments NMT using *retrieval*-based models, retrieving sentence pairs from the training corpus that are most similar to the sentence that we want to translate, and then using these to bias the NMT model.[1] These methods – reminiscent of translation memory (Utiyama et al., 2011) or example-based translation (Nagao, 1984; Grefenstette, 1999) – are effective because they augment the parametric NMT model with a non-parametric translation memory that allows for increased capacity to measure features of the target technical terms or domain-specific words. Currently there are two main approaches to doing so. Li et al. (2016) and Farajian et al. (2017) use the retrieved sentence pairs to fine tune the parameters of the NMT model which is pre-trained on the whole training corpus. Gu et al. (2017) uses the retrieved sentence pairs as additional inputs to the NMT model to help NMT in translating the input sen-

[1] Note that there are existing retrieval-based methods for phrase-based and hierarchical phrase-based translation (Lopez, 2007; Germann, 2015). However, these methods do not improve translation quality but rather aim to improve the efficiency of the translation models.

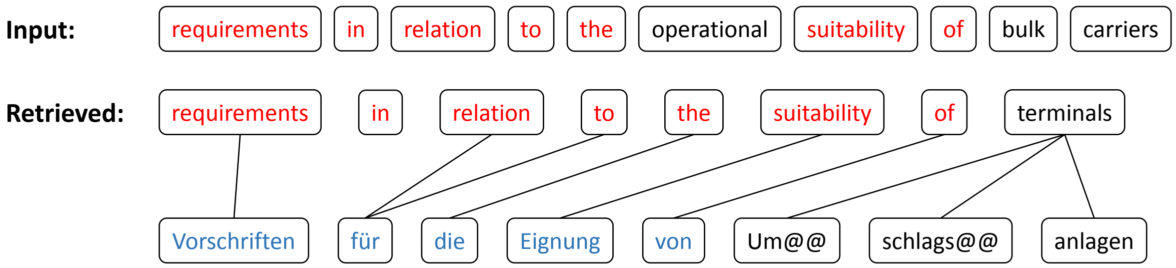

Figure 1: A word-aligned sentence pair retrieved for an input sentence. Red words are unedited words obtained by computing the edit distance between the input sentence and the retrieved source sentence. The blue part of the retrieved target sentence is collected as translation pieces for the input sentence. The target word "Umschlagsanlagen" is split into "Um@@", "schlags@@" and "anlagen" by byte pair encoding.

tence. While both of these paradigms have been proven effective, they both add significant complexity and computational/memory cost to the decoding process, and also to the training procedure. The first requires the running of several training iterations and rolling back of the model, which is costly at test time, and the second requires entirely changing the model structure which requires training the model separately, and also increases test-time computational cost by adding additional encoders.

In this paper, we propose a simple and efficient model for using retrieved sentence pairs to guide an existing NMT model at test time. Specifically, the model collects n-grams occurring in the retrieved target sentences that also match words that overlap between the input and retrieved source sentences, which we will refer to as "translation pieces" (e.g., in Figure 1, the blue part of the retrieved target sentence is collected as translation pieces for the input sentence). The method then calculates a pseudo-probability score for each of the retrieved example sentence pairs and weights the translation pieces according to this value. Finally, we up-weight NMT outputs that contain the collected translation pieces. Unlike the previous methods, this requires no change of the underlying NMT model and no updating of the NMT parameters, making it both simple and efficient to apply at test time.

We show our method improved NMT translation results up to 6 BLEU points on three translation tasks and caused little increase in the translation time. Further, we find that accuracies are comparable with the model of Gu et al. (2017), despite being significantly simpler to implement and faster at test time.

2 Attentional NMT

Our baseline NMT model is similar to the attentional model of Bahdanau et al. (2014), which includes an encoder, a decoder and an attention (alignment) model. Given a source sentence $X = \{x_1, ..., x_L\}$, the encoder learns an annotation $h_i = \left[\vec{h}_i; \overleftarrow{h}_i\right]$ for x_i using a bi-directional recurrent neural network.

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

$$P_{NMT}\left(y_t|y_1^{t-1}, X\right) = softmax\left(g\left(y_{t-1}, z_t, c_t\right)\right) \tag{1}$$

where z_t is a decoding state for time step t, computed by,

$$z_t = f\left(z_{t-1}, y_{t-1}, c_t\right) \tag{2}$$

c_t is a source representation for time t, calculated as,

$$c_t = \sum_{i=1}^{L} \alpha_{t,i} \cdot h_i \tag{3}$$

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

$$\alpha_{t,i} = \frac{\exp\left(a\left(z_{t-1}, h_i\right)\right)}{\sum_{j=1}^{L} \exp\left(a\left(z_{t-1}, h_j\right)\right)} \tag{4}$$

The standard decoding algorithm for NMT is beam search. That is, at each time step t, we keep n-best hypotheses. The probability of a complete

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

hypothesis is computed as,

$$\log P_{NMT}(Y|X) = \sum_{t=1}^{|Y|} \log P_{NMT}\left(y_t | y_1^{t-1}, X\right) \tag{5}$$

Finally, the translation score is normalized by sentence length to avoid too short outputs.

$$\log S_{NMT}(Y|X) = \frac{\log P_{NMT}(Y|X)}{|Y|} \tag{6}$$

3 Guiding NMT with Translation Pieces

This section describes our approach, which mainly consists of two parts:

1. retrieving candidate translation pieces from a parallel corpus for the new source sentence that we want to translate, and then

2. using the collected translation pieces to guide an existing NMT model while translating this new sentence.

At training time, we first prepare the parallel corpus that will form our database used in the retrieval of the translation pieces. Conceivably, it could be possible to use a different corpus for translation piece retrieval and NMT training, for example when using a separate corpus for domain adaptation, but for simplicity in this work we use the same corpus that was used in NMT training. As pre-processing, we use an off-the-shelf word aligner to learn word alignments for the parallel training corpus.

3.1 Retrieving Translation Pieces

At test time we are given an input sentence X. For this X, we first use the off-the-shelf search engine Lucene to search the word-aligned parallel training corpus and retrieve M source sentences $\{X^m : 1 \leq m \leq M\}$ that are similar to X. Y^m indicates the target sentence that corresponds to source sentence X^m and $\mathcal{A}^m$ is word alignments between X^m and Y^m.

For each retrieved source sentence X^m, we compute its edit distance with X as $d(X, X^m)$ using dynamic programming. We record the unedited words in X^m as $\mathcal{W}^m$, and also note the words in the target sentence Y^m that correspond to source words in $\mathcal{W}^m$, which we can presume are words that will be more likely to appear in the translated sentence for X. According to Algorithm 1, we collect n-grams (up to 4-grams) from the retrieved target sentence Y^m as possible translation pieces G_X^m for X, using word-level alignments to select n-grams that are related to X and discard n-grams that are not related to X. The final translation pieces G_X collected for X are computed as,[3]

n-grams	G_X^m
Vorschriften für die Eignung	Yes
die Eignung von	Yes
von Um@@ schlags@@ anlagen	No
Um@@ schlags@@ anlagen	No

Table 1: Examples of the collected translation pieces.

$$G_X = \bigcup_{m=1}^{M} G_X^m \tag{7}$$

Table 1 shows a few n-gram examples contained in the retrieved target sentence in Figure 1 and whether they are included in G_X^m or not. Because the retrieved source sentence in Figure 1 is highly similar with the input sentence, the translation pieces collected from its target side are highly likely to be correct translation pieces of the input sentence. However, when a retrieved source sentence is not very similar with the input sentence (e.g. only one or two words match), the translation pieces collected from its target side will be less likely to be correct translation pieces for the input sentence.

We compute a score for each $u \in G_X$ to measure how likely it is a correct translation piece for X based on sentence similarity between the retrieved source sentences and the input sentence as following,

$$\begin{aligned} & S\left(u, X, \bigcup_{m=1}^{M} \{(X^m, G_X^m)\}\right) \\ & = \max_{1 \leq m \leq M \wedge u \in G_X^m} simi(X, X^m) \end{aligned} \tag{8}$$

where $simi(X, X^m)$ is the sentence similarity computed as following (Gu et al., 2017),

$$simi(X, X^m) = 1 - \frac{d(X, X^m)}{\max(|X|, |X^m|)} \tag{9}$$

[3] Note that the extracted translation pieces are target phrases, but the target words contained in one extracted translation piece may be aligned to discontiguous source words, which is different from how phrase-based translation extracts phrase-based translation rules.

Algorithm 1 Collecting Translation Pieces

```
Require: $X = x_1^L$, $X^m = k_1^{L'}$, $Y^m = v_1^{L''}$, $\mathcal{A}^m$, $\mathcal{W}^m$
Ensure: $G_X^m$
  $G_X^m = \emptyset$
  for $i = 1$ to $L''$ do
    for $j = i$ to $L''$ do
      if $j - i = 4$ then
        break
      if $\exists p : (p, j) \in \mathcal{A}^m \wedge p \notin \mathcal{W}^m$ then
        break
      add $v_i^j$ into $G_X^m$
```

Algorithm 2 Guiding NMT by Translation Pieces

```
Require: Output layer $\log P_{NMT}\left(y_t|y_1^{t-1}, X\right)$, $\mathcal{L}_X, \mathcal{D}_X$
Ensure: Updated output layer
  for $u$ in $\mathcal{L}_X$ do
    $\log P_{NMT}\left(u|y_1^{t-1}, X\right) += \lambda \mathcal{D}_X\left(u\right)$
    for $i = 1$ to $3$ do
      if $t - i < 1$ then
        break
      if $y_{t-i}^{t-1}, u \notin \mathcal{D}_X$ then
        break
      $\log P_{NMT}\left(u|y_1^{t-1}, X\right) += \lambda \mathcal{D}_X\left(y_{t-i}^{t-1}, u\right)$
```

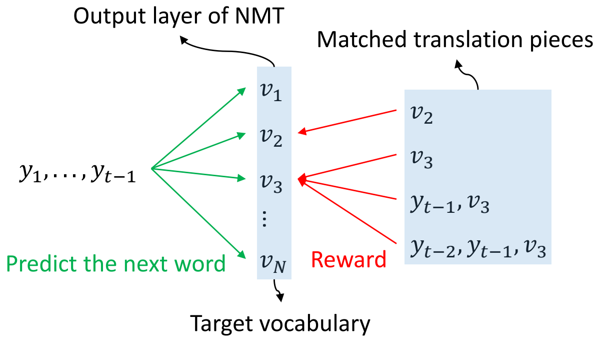

Figure 2: A simple demonstration of adding rewards for matched translation pieces into the NMT output layer.

3.2 Guiding NMT with Retrieved Translation Pieces

In the next phase, we use our NMT system to translate the input sentence. Inspired by Stahlberg et al. (2017) which rewards n-grams from syntactic translation lattices during NMT decoding, we add an additional reward for n-grams that occur in the collected translation pieces. That is, as shown in Figure 2, at each time step t, we update the probabilities over the output vocabulary and increase the probabilities of those that result in matched n-grams according to

$$\begin{aligned}\log S_{NMT\_updated}\left(y_t|y_1^{t-1}, X\right) \\ = \log P_{NMT}\left(y_t|y_1^{t-1}, X\right) + \\ \lambda \sum_{n=1}^{4} \delta\left(y_{t-n+1}^{t}, X, \bigcup_{m=1}^{M} \{(X^m, G_X^m)\}\right),\end{aligned} \tag{10}$$

where λ can be tuned on the development set and $\delta\left(\cdot\right)$ is computed as Equation 8 if $y_{t-n+1}^{t} \in G_X$, otherwise $\delta\left(\cdot\right) = 0$.

To implement our method, we use a dictionary $\mathcal{D}_X$ to store translation pieces G_X and their scores for each input sentence X. At each time step t, we update the output layer probabilities by checking $\mathcal{D}_X$. However, it is inefficient to traverse all target words in the vocabulary and check whether they belong to G_X or not, because the vocabulary size is large. Instead, we only traverse target words that belong to G_X and update the corresponding output probabilities as shown in Algorithm 2. Here, $\mathcal{L}_X$ is a list that stores 1-grams contained in G_X.[4]

As we can see, our method only up-weights NMT outputs that match the retrieved translation pieces in the NMT output layer. In contrast, Li et al. (2016) and Farajian et al. (2017) use the retrieved sentence pairs to run additional training iterations and fine tune the NMT parameters for each input sentence; Gu et al. (2017) runs the NMT model for each retrieved sentence pair to obtain the NMT encoding and decoding information of the retrieved sentences as key-value memory to guide NMT for translating the new input sentence. Compared to their methods, our method adds little computational/memory cost and is simple to implement.

4 Experiment

4.1 Settings

Following Gu et al. (2017), we use version 3.0 of the JRC-Acquis corpus for our translation experiments. The JRC-Acquis corpus contains the total body of European Union (EU) law applicable in the EU Member States. It can be used as a narrow domain to test the effectiveness of our proposed method. We did translation experiments on three

[4] Note that our method does not introduce new states during decoding, because the output layer probabilities are simply updated based on history words and the next word.

		en-de		en-fr		en-es	
		BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
dev	NMT	44.08	36.69	57.26	43.51	55.76	42.53
	Ours	50.81	39.50	62.60	45.83	60.51	44.58
test	NMT	43.76	36.57	57.67	43.66	55.78	42.55
	Ours	50.15	39.18	63.27	46.24	60.54	44.64

Table 2: Translation results.

	en-de	en-fr	en-es
TRAIN	674K	665K	663K
DEV	1,636	1,733	1,662
TEST	1,689	1,710	1,696
Average Length	31	29	29

Table 3: Data sets. The last line is the average length of English sentences.

directions: English-to-German (en-de), English-to-French (en-fr) and English-to-Spanish (en-es).

We cleaned the data by removing repeated sentences and used the `train-truecaser.perl` script from Moses (Koehn et al., 2007) to truecase the corpus. Then we selected 2000 sentence pairs as development and test sets, respectively. The rest was used as the training set. We removed sentences longer than 80 and 100 from the training and development/test sets respectively. The final numbers of sentence pairs contained in the training, development and test sets are shown in Table 3.[5] We applied *byte pair encoding* (Sennrich et al., 2016b) and set the vocabulary size to be 20K.

For translation piece collection, we use GIZA++ (Och and Ney, 2003) and the `grow-diag-final-and` heuristic (Koehn et al., 2003) to obtain symmetric word alignments for the training set.

We trained an attentional NMT model as our baseline system. The settings for NMT are shown in Table 4. We also compared our method with the search engine guided NMT model (SGNMT, Gu et al. (2017)) in Section 4.5.

Word embedding	512
GRU dimension	1024
Optimizer	adam
Initial learning rate	0.0001
Beam size	5

Table 4: NMT settings.

		en-de	en-fr	en-es
dev	NMT	1.000	0.990	0.997
	Ours	1.005	0.991	1.001
test	NMT	0.995	0.990	0.990
	Ours	1.004	0.989	0.993

Table 5: Ratio of translation length to reference length.

For each input sentence, we retrieved 100 sentence pairs from the training set using Lucene as our preliminary setting. We analyze the influence of the retrieval size in Section 4.4. The weights of translation pieces used in Equation 10 are tuned on the development set for different language pairs, resulting in weights of 1.5 for en-de and en-fr, and a weight of 1 for en-es.

4.2 Results

Table 2 shows the main experimental results. We can see that our method outperformed the baseline NMT system up to 6 BLEU points. As large BLEU gains in neural MT can also often be attributed to changes in output length, we examined the length (Table 5) and found that it did not influence the translation length significantly.

In addition, it is of interest whether how well the retrieved sentences match the input influences the search results. We measure the similarity between a test sentence X and the training corpus D_{train} by computing the sentence similarities between X and the retrieved source sentences as

$$simi\left(X, D_{train}\right) = \max_{1 \leq m \leq M} simi\left(X, X^m\right). \tag{11}$$

The similarity between the test set D_{test} and the training corpus D_{train} is measured as,

$$simi\left(D_{test}, D_{train}\right) = \frac{\sum_{X \in D_{test}} simi\left(X, D_{train}\right)}{|D_{test}|} \tag{12}$$

Our analysis demonstrated that, expectedly, the performance of our method is highly influenced by the similarity between the test set and the training set. We divided sentences in the test set into two

[5] We put the datasets used in our experiments on Github https://github.com/jingyiz/Data-sampled-preprocessed

	whole	half-H	half-L
en-de	0.56	0.80	0.32
en-fr	0.57	0.81	0.33
en-es	0.57	0.81	0.32

Table 6: Similarities between the training set and the whole/divided test sets.

		whole	half-H	half-L
en-de	NMT	43.76	60.93	32.25
	Ours	50.15	73.26	34.28
en-fr	NMT	57.67	72.64	47.38
	Ours	63.27	82.76	49.81
en-es	NMT	55.78	69.32	46.26
	Ours	60.54	78.37	47.93

Table 7: Translation results (BLEU) for the whole/divided test sets.

parts: half has higher similarities with the training corpus (half-H) and half has lower similarities with the training corpus (half-L). Table 6 shows the similarity between the training corpus and the whole/divided test sets. Table 7 shows translation results for the whole/divided test sets. As we can see, NMT generally achieved better BLEU scores for half-H and our method improved BLEU scores for half-H much more significantly than for half-L, which shows our method can be quite useful for narrow domains where similar sentences can be found.

We also tried our method on WMT 2017 English-to-German News translation task. However, we did not achieve significant improvements over the baseline attentional NMT model, likely because the test set and the training set for the WMT task have a relatively low similarity as shown in Table 8 and hence few useful translation pieces can be retrieved for our method. In contrast, the JRC-Acquis corpus provides test sentences that have much higher similarities with the training set, i.e., much more and longer translation pieces exist.

To demonstrate how the retrieved translation pieces help NMT to generate appropriate outputs, Figure 3 shows an input sentence with reference, the retrieved sentence pair with the highest sentence similarity and outputs by different systems for this input sentence with detailed scores: log NMT probabilities for each target word in T_1 and T_2; scores for matched translation pieces contained in T_1 and T_2. As we can see, NMT assigns higher probabilities to the incorrect translation T_1, even though the retrieved sentence pair whose source side is very similar with the input sentence was used for NMT training.

	WMT		JRC-Acquis	
Similarity	Sent	Percent	Sent	Percent
$[0, 0.1)$	0	0%	4	0.2%
$[0.1, 0.2)$	415	13.8%	141	8.3%
$[0.2, 0.3)$	1399	46.5%	238	14.0%
$[0.3, 0.4)$	740	24.6%	194	11.4%
$[0.4, 0.5)$	281	9.3%	154	9.1%
$[0.5, 0.6)$	113	3.7%	156	9.2%
$[0.6, 0.7)$	29	0.9%	157	9.2%
$[0.7, 0.8)$	10	0.3%	156	9.2%
$[0.8, 0.9)$	10	0.3%	252	14.9%
$[0.9, 1)$	0	0%	237	14.0%
1	7	0.2%	0	0%

Table 8: Statistics for similarities between each test sentence and the training set as computed by Equation 11 for the WMT 2017 en-de task (3004 sentences) and our JRC-Acquis en-de task (1689 sentences).

		en-de	en-fr	en-es
dev	NMT	44.08	57.26	55.76
	Ours	50.81	62.60	60.51
	1/0 reward	47.70	61.15	58.92
test	NMT	43.76	57.67	55.78
	Ours	50.15	63.27	60.54
	1/0 reward	47.13	62.14	58.66

Table 9: Translation results (BLEU) of 1/0 reward.

However, T_2 contains more and longer translation pieces with higher scores. The five translation pieces contained only in T_2 are collected from the retrieved sentence pair shown in Figure 3, which has high sentence similarity with the input sentence. The three translation pieces contained only in T_1 are also translation pieces collected for the input sentence, but have lower scores, because they are collected from sentence pairs with lower similarities with the input sentence. This shows that computing scores for translation pieces based on sentence similarities is important for the performance of our method. If we assign score 1 to all translation pieces contained in G_X, i.e., use 1/0 reward for translation pieces and non-translation pieces, then the performance of our method decreased significantly as shown in Table 9, but still outperformed the NMT baseline significantly.

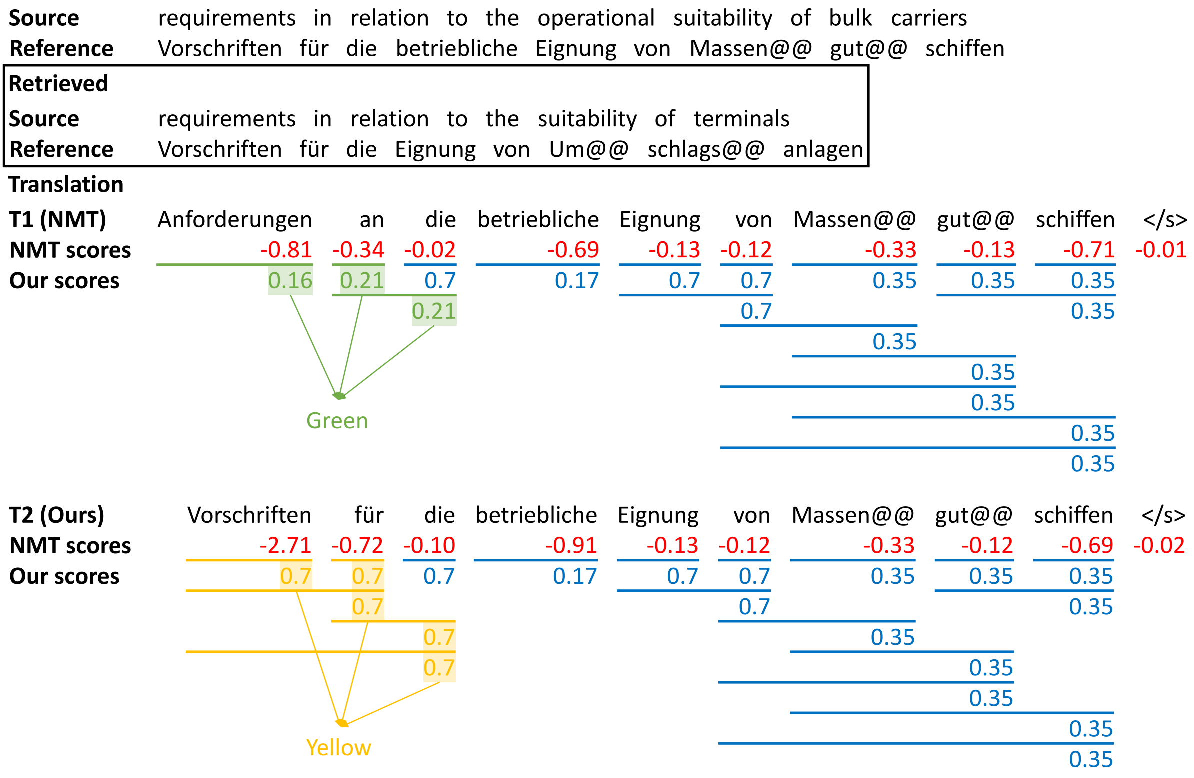
Source requirements in relation to the operational suitability of bulk carriers
Reference Vorschriften für die betriebliche Eignung von Massen@@ gut@@ schiffen
Retrieved
Source requirements in relation to the suitability of terminals
Reference Vorschriften für die Eignung von Um@@ schlags@@ anlagen
Translation
T1 (NMT) Anforderungen an die betriebliche Eignung von Massen@@ gut@@ schiffen </s>
NMT scores -0.81 -0.34 -0.02 -0.69 -0.13 -0.12 -0.33 -0.13 -0.71 -0.01
Our scores 0.16 0.21 0.7 0.17 0.7 0.7 0.35 0.35 0.35
0.21 0.7 0.35
0.35
0.35
0.35
0.35
0.35
Green
T2 (Ours) Vorschriften für die betriebliche Eignung von Massen@@ gut@@ schiffen </s>
NMT scores -2.71 -0.72 -0.10 -0.91 -0.13 -0.12 -0.33 -0.12 -0.69 -0.02
Our scores 0.7 0.7 0.7 0.17 0.7 0.7 0.35 0.35 0.35
0.7 0.7 0.35
0.7 0.35
0.7 0.35
0.35
0.35
0.35
0.35
Yellow

Figure 3: Translation examples. Red scores are log NMT probabilities. Green, yellow and blue scores are scores of matched translation pieces contained only in T_1, contained only in T_2, contained in both T_1 and T_2, respectively.

γ		0	1	2	5	10	20	50	100
en-de	NMT	5834	3193	1988	1196	717	370	157	75
	Ours	5843	5433	3153	1690	933	458	193	86
	Ratio (Ours/NMT)	1.00	1.70	1.58	1.41	1.30	1.23	1.22	1.14
en-fr	NMT	6983	3743	2637	1563	812	493	210	118
	Ours	7058	5443	3584	1919	968	581	214	134
	Ratio (Ours/NMT)	1.01	1.45	1.35	1.22	1.19	1.17	1.01	1.13
en-es	NMT	6500	3430	2292	1346	772	437	182	95
	Ours	6516	4589	2970	1652	895	500	196	97
	Ratio (Ours/NMT)	1.00	1.33	1.29	1.22	1.15	1.14	1.07	1.02

Table 10: $Count_{\gamma}$

4.3 Infrequent n-grams

The basic idea of our method is rewarding n-grams that occur in the training set during NMT decoding. We found our method is especially useful to help the translation for infrequent n-grams. First, we count how many times a target n-gram u occurs in the training set D_{train} as,

$$Occur\left(u\right) = \left|\left\{Y : \langle X, Y\rangle \in D_{train} \wedge u \in uniq\left(Y\right)\right\}\right| \quad (13)$$

where $uniq\left(Y\right)$ is the set of uniq n-grams (up to 4-grams) contained in Y.

Given system outputs $\left\{Z^k : 1 \leq k \leq K\right\}$ for the test set $\left\{X^k : 1 \leq k \leq K\right\}$ with reference $\left\{Y^k : 1 \leq k \leq K\right\}$, we count the number of correctly translated n-grams that occur γ times in the training set as,

$$Count_{\gamma} = \sum_{k=1}^{K} \left|\psi\left(\gamma, Z^k, Y^k\right)\right| \quad (14)$$

where

$$\psi\left(\gamma, Z^k, Y^k\right) = \left\{u : u \in \left(uniq\left(Z^k\right) \cap uniq\left(Y^k\right)\right) \wedge Occur\left(u\right) = \gamma\right\} \quad (15)$$

Table 10 shows $Count_{\gamma}$ for different system outputs. As we can see, our method helped little for the translation of n-grams that do not occur

	en-de	en-fr	en-es
Base NMT decoding	0.215	0.224	0.227
Search engine retrieval	0.016	0.017	0.016
TP collection	0.521	0.522	0.520
Our NMT decoding	0.306	0.287	0.289

Table 11: Translation time (seconds).

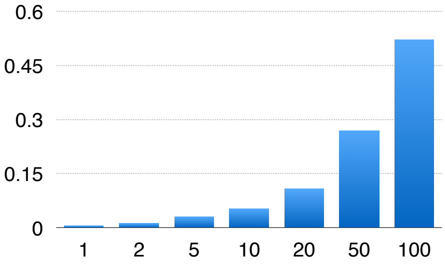

Figure 4: Translation piece collection time (seconds) with different search engine retrieval sizes.

in the training set, which is reasonable because we only reward n-grams that occur in the training set. However, our method helped significantly for the translation of n-grams that do occur in the training set but are infrequent (occur less than 5 times). As the frequency of n-grams increases, the improvement caused by our method decreased. We analyze that the reason why our method is especially helpful for infrequent n-grams is that NMT is trained on the whole training corpus for maximum likelihood and tends to generate more frequent n-grams while our method computes scores for the collected translation pieces based on sentence similarities and does not prefer more frequent n-grams.

4.4 Computational Considerations

Our method only collects translation pieces to help NMT for translating a new sentence and does not influence the training process of NMT. Therefore, our method does not increase the NMT training time. Table 11 shows the average time needed for translating one input sentence in the development set in our experiments. The search engine retrieval and translation piece (TP) collection time is computed on a 3.47GHz Intel Xeon X5690 machine using one CPU. The NMT decoding time is computed using one GPU GeForce GTX 1080.

As we can see, the search engine retrieval time is negligible and the increase of NMT decoding time caused by our method is also small. However, collecting translation pieces needed considerable time, although our implementation was in Python and could potentially be significantly faster in a more efficient programming language. The translation piece collection step mainly consists of two parts: computing the edit distances between the input sentence and the retrieved source sentences using dynamic programming with time complexity $O\left(n^2\right)$; collecting translation pieces using Algorithm 1 with time complexity $O\left(4n\right)$.

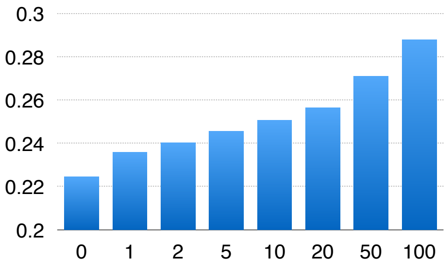

Figure 5: NMT decoding time (seconds) with different search engine retrieval sizes.

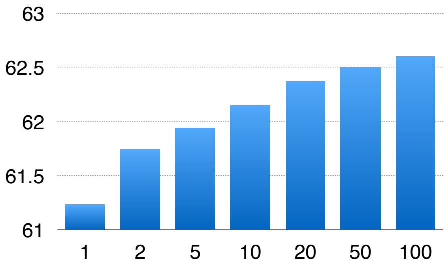

Figure 6: Translation results (BLEU) with different search engine retrieval sizes.

We changed the size of sentence pairs retrieved by the search engine and analyze its influence on translation performance and time. Figure 4, 5 and 6 show the translation piece collection time, the NMT decoding time and translation BLEU scores with different search engine retrieval sizes for the en-fr task. As we can see, as the number of retrieved sentences decreased, the time needed by translation piece collection decreased significantly, the translation performance decreased much less significantly and the NMT decoding time is further reduced. In our experiments, 10 is a good setting for the retrieval size, which gave significant BLEU score improvements and caused little increase in the total translation time compared to the NMT baseline.

		en-de	en-fr	en-es
dev	$\text{NMT}_{reported}$	44.94	58.95	50.54
	$\text{SGNMT}_{reported}$	49.26	64.16	**57.62**
	NMT	45.18	59.08	50.71
	Ours	**50.61**	**65.03**	57.49
test	$\text{NMT}_{reported}$	43.98	59.42	50.48
	$\text{SGNMT}_{reported}$	48.80	64.60	**57.27**
	NMT	44.21	59.43	50.61
	Ours	**50.36**	**65.69**	57.11

Table 12: Comparison with SGNMT.

4.5 Comparison with SGNMT

We compared our method with the search engine guided NMT (SGNMT) model (Gu et al., 2017). We got their preprocessed datasets and tested our method on their datasets, in order to fairly compare our method with their reported BLEU scores.[6] Table 12 shows the results of their method and our method with the same settings for the baseline NMT system. As we can see, our method generally outperformed their method on the three translation tasks.

Considering the computational complexity, their method also performs search engine retrieval for each input sentence and computes the edit distance between the input sentence and the retrieved source sentences as our method. In addition, their method runs the NMT model for each retrieved sentence pair to obtain the NMT encoding and decoding information of the retrieved sentences as key-value memory to guide the NMT model for translating the real input sentence, which changes the NMT model structure and increases both the training-time and test-time computational cost. Specifically, at test time, running the NMT model for one retrieved sentence pair costs the same time as translating the retrieved source sentence with beam size 1. Therefore, as the number of the retrieved sentence pairs increases to the beam size of the baseline NMT model, their method doubles the translation time.

5 Conclusion

This paper presents a simple and effective method that retrieves translation pieces to guide NMT for narrow domains. We first exploit a search engine to retrieve sentence pairs whose source sides are similar with the input sentence, from which we collect and weight translation pieces for the input sentence based on word-level alignments and sentence similarities. Then we use an existing NMT model to translate this input sentence and give an additional bonus to outputs that contain the collected translation pieces. We show our method improved NMT translation results up to 6 BLEU points on three narrow domain translation tasks, caused little increase in the translation time, and compared favorably to another alternative retrieval-based method with respect to accuracy, speed, and simplicity of implementation.

[6] Only BLEU scores are reported in their paper.

Acknowledgments

We thank Jiatao Gu for providing their preprocessed datasets in Section 4.5.

References

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. https://aclweb.org/anthology/D16-1162.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* https://arxiv.org/pdf/1409.0473.pdf.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*. pages 157–168. http://www.aclweb.org/anthology/W17-4716.

Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov, and Shahram Khadivi. 2017. Neural machine translation leveraging phrase-based models in a hybrid search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1411–1420. https://www.aclweb.org/anthology/D17-1148.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*. pages 127–137. http://www.aclweb.org/anthology/W/W17/W17-4713.pdf.

Ulrich Germann. 2015. Sampling phrase tables for the moses statistical machine translation system. *The Prague Bulletin of Mathematical Linguistics* 104(1):39–50. https://ufal.mff.cuni.cz/pbml/104/art-germann.pdf.

Gregory Grefenstette. 1999. The world wide web as a resource for example-based machine translation tasks. In *Proceedings of the ASLIB Conference on Translating and the Computer*. volume 21. http://www.mt-archive.info/Aslib-1999-Grefenstette.pdf.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2017. Search engine guided non-parametric neural machine translation. *arXiv preprint arXiv:1705.07267* https://arxiv.org/pdf/1705.07267.pdf.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with smt features. In *AAAI*. pages 151–157. https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12189/11577.

Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. 2017. Learning to remember rare events. *arXiv preprint arXiv:1703.03129* https://arxiv.org/pdf/1703.03129.pdf.

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. http://www.aclweb.org/anthology/P07-2045.

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. http://www.aclweb.org/anthology/W17-3204.

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. http://www.aclweb.org/anthology/N/N03/N03-1017.pdf.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2016. One sentence one model for neural machine translation. *arXiv preprint arXiv:1609.06490* https://arxiv.org/pdf/1609.06490.pdf.

Adam Lopez. 2007. Hierarchical phrase-based translation with suffix arrays. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. pages 976–985. http://www.aclweb.org/anthology/D/D07/D07-1104.

Makoto Nagao. 1984. A framework of a mechanical translation between japanese and english by analogy principle. *Artificial and human intelligence* pages 351–354. http://www.mt-archive.info/Nagao-1984.pdf.

Toan Q Nguyen and David Chiang. 2017. Improving lexical choice in neural machine translation. *arXiv preprint arXiv:1710.01329* https://arxiv.org/pdf/1710.01329.pdf.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics* 29(1):19–51. http://www.aclweb.org/anthology/J/J03/J03-1002.pdf.

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. http://www.aclweb.org/anthology/W/W16/W16-2323.pdf.

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. http://www.aclweb.org/anthology/P/P16/P16-1162.pdf.

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. http://www.aclweb.org/anthology/E/E17/E17-2058.pdf.

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* https://arxiv.org/pdf/1606.01792.pdf.

Masao Utiyama, Graham Neubig, Takashi Onishi, and Eiichiro Sumita. 2011. Searching translation memories for paraphrases. In *Machine Translation Summit*. volume 13, pages 325–331. http://mt-archive.info/MTS-2011-Utiyama.pdf.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017a. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. pages 560–566. http://aclweb.org/anthology/P17-2089.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017b. Sogou neural machine translation systems for wmt17. In *Proceedings of the Second Conference on Machine Translation*. pages 410–415. http://www.aclweb.org/anthology/W17-4742.

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. http://www.aclweb.org/anthology/P/P17/P17-1139.pdf.

Jian Zhang, Liangyou Li, Andy Way, and Qun Liu. 2016. Topic-informed neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. pages 1807–1817. http://aclweb.org/anthology/C16-1170.