Code-Switching for Enhancing NMT with Pre-Specified Translation

Kai Song[1,2], Yue Zhang[3], Heng Yu[2], Weihua Luo[2], Kun Wang[1], Min Zhang[1]

[1] Soochow University, Suzhou, China
[2] Alibaba DAMO Academy, Hangzhou, China
[3] School of Engineering, Westlake University, Hangzhou, China
songkai.sk@alibaba-inc.com, zhangyue@wias.org.cn
{yuheng.yh,weihua.luowh}@alibaba-inc.com
kwang1994@stu.suda.edu.cn,minzhang@suda.edu.cn

Abstract

Leveraging user-provided translation to constrain NMT has practical significance. Existing methods can be classified into two main categories, namely the use of placeholder tags for lexicon words and the use of hard constraints during decoding. Both methods can hurt translation fidelity for various reasons. We investigate a data augmentation method, making code-switched training data by replacing source phrases with their target translations. Our method does not change the NMT model or decoding algorithm, allowing the model to learn lexicon translations by copying source-side target words. Extensive experiments show that our method achieves consistent improvements over existing approaches, improving translation of constrained words without hurting unconstrained words.

1 Introduction

One important research question in domain-specific machine translation (Luong and Manning, 2015) is how to impose translation constraints (Crego et al., 2016; Hokamp and Liu, 2017; Post and Vilar, 2018). As shown in Figure 1 (a), the word "breadboard" can be translated into "切面包板 (a wooden board that is used to cut bread on)" in the food domain, but "电路板 (a construction base for prototyping of electronics)" in the electronic domain. To enhance translation quality, a lexicon can be leveraged for domain-specific or user-provided words (Arthur et al., 2016; Hasler et al., 2018). We investigate the method of leveraging pre-specified translation for NMT using such a lexicon.

For leveraging pre-specified translation, one existing approach uses *placeholder tags* to substitute named entities (Crego et al., 2016; Li et al., 2016; Wang et al., 2017b) or rare words (Luong et al.,
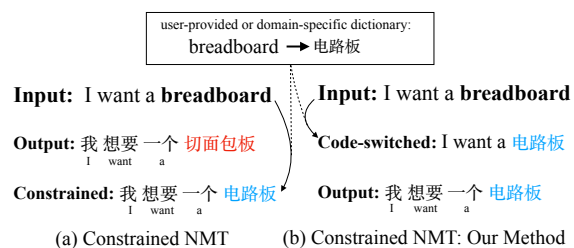

Figure 1: Constrained NMT

2014) on both the source and target sides during training, so that a model can translate such words by learning to translate placeholder tags. For example, the i-th named entity in the source sentence is replaced with "tag_i", as well as its corresponding translation in the target side. Placeholder tags in the output are replaced with pre-specified translation as a post-processing step. One disadvantage of this approach, however, is that the meaning of the original words in the pre-specified translation is not fully retained, which can be harmful to both adequacy and fluency of the output.

Another approach (Hokamp and Liu, 2017; Post and Vilar, 2018) imposes pre-specified translation via *lexical constraints*, making sure such constraints are satisfied by modifying NMT decoding. This method ensures that pre-specified translations appear in the output. A problem of this method is that it does not explicitly explore the correlation between pre-specified translations and their corresponding source words during decoding, and thus can hurt translation fidelity (Hasler et al., 2018). There is not a mechanism that allows the model to *learn* constraint translations during training, which the placeholder method allows.

We investigate a novel method based on *data augmentation*, which combines the advantages of both methods above. The idea is to construct synthetic parallel sentences from the original paral-

lel training data. The synthetic sentence pairs resemble code-switched source sentences and their translations, where certain source words are replaced with their corresponding target translations. The motivation is to make the model learn to "translate" embedded pre-specified translations by copying them from the modified source. During decoding, the source is similarly modified as a pre-processing step. As shown in Figure 1 (b), translation is executed over the code-switched source, without further constraints or post-processing.

In contrast to the placeholder method, our method keeps lexical semantic information (i.e. target words v.s. placeholder tags) in the source, which can lead to more adequate translations. Compared with the lexical constraint method, pre-specified translation is *learned* because such information is available both in training and decoding. As a data augmentation method, it can be used on any NMT architecture. In addition, our method enables the model to translate code-switched source sentences, and preserve its strength in translating un-replaced sentences.

To further strengthen copying, we propose two model-level adjustments: First, we share target-side embeddings with source-side target words, so that target vocabulary words have a unique embedding in the NMT system. Second, we integrate pointer network (Vinyals et al., 2015; Gulcehre et al., 2016; Gu et al., 2016; See et al., 2017) into the decoder. The copy mechanism was firstly proposed to copy source words. In our method, it is further used to copy source-side *target* words.

Results on large scale English-to-Russian (En-Ru) and Chinese-to-English (Ch-En) tasks show that our method outperforms both placeholder and lexical constraint methods over a state-of-the-art Transformer (Vaswani et al., 2017) model on various test sets across different domains. We also show that shared embedding and pointer network can lead to more successful applications of the copying mechanism. We release four high-quality En-Ru e-commerce test sets translated by Russian language experts, totalling 7169 sentences with an average length of 21[1].

2 Related Work

Using placeholders. Luong et al. (2014) use *annotated unk tags* to present the *unk* symbols in

training corpora, where the correspondence between source and target *unk* symbols are obtained from word alignment (Brown et al., 1993). Output *unk tags* are replaced through a post-processing stage by looking up a pre-specified dictionary or copying the corresponding source word. Crego et al. (2016) extended *unk tags* symbol to specific symbols that can present name entities. Wang et al. (2017b) and Li et al. (2016) use a similar method. This method is limited when constrain NMT with pre-specified translations consisting of more general words, due to the loss of word meaning when representing them with placeholder tags. In contrast to their work, word meaning is fully kept in modified source in our work.

Lexical constraints. Hokamp and Liu (2017) propose an altered beam search algorithm, namely grid beam search, which takes target-side pre-specified translations as lexical constraints during beam search. A potential problem of this method is that translation fidelity is not specifically considered, since there is no indication of a matching source of each pre-specific translation. In addition, decoding speed is significantly reduced (Post and Vilar, 2018). Hasler et al. (2018) use alignment to gain target-side constraints' corresponding source words, simultaneously use finite-state machines and multi-stack (Anderson et al., 2016) decoding to guide beam search. Post and Vilar (2018) give a fast version of Hokamp and Liu (2017), which limits the decoding complexity linearly by altering the beam search algorithm through dynamic beam allocation.

In contrast to their methods, our method does not make changes to the decoder, and therefore decoding speed remains unchanged. Translation fidelity of pre-specified source words is achieved through a combination of training and decoding procedure, where replaced source-side words still contain their target-side meaning. As a soft method of inserting pre-specified translation, our method does not guarantee that all lexical constraints are satisfied during decoding, but has better overall translation quality compared to their method.

Using probabilistic lexicons. Aiming at making use of one-to-many phrasal translations, the following work is remotely related to our work. Tang et al. (2016) use a phrase memory to provide extra information for their NMT encoder, dynamically switching between word generation and

---

[1]To best of our knowledge, this is the first public e-commerce test set.

phrase generation during decoding. Wang et al. (2017a) use SMT to recommend prediction for NMT, which contains not only translation operations of a SMT phrase table, but also alignment information and coverage information. Arthur et al. (2016) incorporate discrete lexicons by converting lexicon probabilities into predictive probabilities and linearly interpolating them with NMT probability distributions.

Our method is similar in the sense that external translations of source phrases are leveraged. However, their tasks are different. In particular, these methods regard one-to-many translation lexicons as a *suggestion*. In contrast, our task aims to *constrain* NMT translation through **one-to-one** pre-specified translations. Lexical translations can be used to generate code-switched source sentences during training, but we do not modify NMT models by integrating translation lexicons. In addition, our data augmentation method is more flexible, because it is model-free.

Alkhouli et al. (2018) simulate a dictionary-guided translation task to evaluate NMT's alignment extraction. A one-to-one word translation dictionary is used to guide NMT decoding. In their method, a dictionary entry is limited to only one word on both the source and target sides. In addition, a pre-specified translation can come into effect only if the corresponding source-side word is successfully aligned during decoding.

On translating named entities, Currey et al. (2017) augment the training data by copying target-side sentences to the source-side, resulting in augmented training corpora where the source and the target sides contain identical sentences. The augmented data is shown to improve translation performance, especially for proper nouns and other words that are identical in the source and target languages.

3 Data augmentation

Our method is based on *data augmentation*. During training, augmented data are generated by replacing source words or phrases directly with their corresponding target translations. The motivation is to sample as many code-switched translation pairs as possible. During decoding, given pre-specified translations, the source sentence is modified by replacing phrases with their pre-specified translations, so that the trained model can directly copy embedded target translations in the output.

3.1 Training

Given a bilingual training corpus, we sample augmented sentence pairs by leveraging a SMT phrase table, which can be trained over the same bilingual corpus or a different large corpus. We extract source-target phrase pairs[2] from the phrase table, replacing source-side phrases of source sentences using the following sampling steps:

1. Indexing between source-target phrase pairs and training sentences: (a) For each source-target phrase pair, we record all the matching bilingual sentences that contain both the source and target. Word alignment can be used to ensure the phrase pairs that are mutual translation. (b) We also sample bilingual sentences that match two source-target phrase pairs. In particular, given a combination of two phrase pairs, we index bilingual sentences that match both simultaneously.

2. Sampling: (a) For each source-target phrase pair, we keep at most k_1 randomly selected matching sentences. The source-side phrase is replaced with its target-side translation. (b) For each combination of two source-target phrase pairs, we randomly sample at most k_2 matching sentences. Both source-side matching phrases are replaced with their target translations.[3]

The sampled training data is added to the original training data to form a final set of training sentences.

3.2 Decoding

We impose target-side pre-specified translations to the source by replacing source phrases with their translations. Lexicons are defined in the form of one-to-one source-target phrase pairs. Different from training, the number of replaced phrases in a source sentence is not necessarily restricted to one or two, which will be discussed in Section 5.5. In practice, pre-specified translations can be provided by customers or through user feedback, which contains one identified translation for specified source segment.

4 Model

Transformer (Vaswani et al., 2017) uses self-attention network for both encoding and decod-

---

[2]Source-side phrase is at most trigram.
[3]We set $k_1 = 100, k_2 = 30$ empirically.

ing. The encoder is composed of n stacked neural layers. For time step i in layer j, the hidden state $h_{i,j}$ is calculated by employing self-attention over the hidden states in layer $j-1$, which are $\{h_{1,j-1}, h_{2,j-1}, ..., h_{m,j-1}\}$, where m is the number of source-side words.

In particular, $h_{i,j}$ is calculated as follows: First, a self-attention sub-layer is employed to encode the context. Then *attention weights* are computed as scaled dot product between the current query $h_{i,j-1}$ and all keys $\{h_{1,j-1}, h_{2,j-1}, ..., h_{m,j-1}\}$, normalized with a softmax function. After that, the context vector is represented as *weighted sum* of the values projected from hidden states in the previous layer, which are $\{h_{1,j-1}, h_{2,j-1}, ..., h_{m,j-1}\}$. The hidden state in the previous layer and the context vector are then connected by residual connection, followed by a layer normalization function (Ba et al., 2016), to produce a candidate hidden state $h'_{i,j}$. Finally, another sub-layer including a feed-forward network (FFN) layer, followed by another residual connection and layer normalization, are used to obtain the hidden state $h_{i,j}$.

In consideration of translation quality, multi-head attention is used instead of single-head attention as mentioned above, positional encoding is also used to compensate the missing of position information in this model.

The decoder is also composed of n stacked layers. For time step t in layer j, a self-attention sub-layer of hidden state $s_{t,j}$ is calculated by employing self-attention mechanism over hidden states in previous target layer, which are $\{s_{1,j-1}, s_{2,j-1}, ..., s_{t-1,j-1}\}$, resulting in candidate hidden state $s'_{t,j}$. Then, a second target-to-source sub-layer of hidden state $s_{t,j}$ is inserted above the target self-attention sub-layer. In particular, the queries(Q) are projected from $s'_{t,j}$, and the keys(K) and values(V) are projected from the source hidden states in the last layer of encoder, which are $\{h_{1,n}, h_{2,n}, ..., h_{m,n}\}$. The output state is another candidate hidden state $s''_{t,j}$. Finally, a last feed-forward sub-layer of hidden state $s_{t,j}$ is calculated by employing self-attention over $s''_{t,j}$.

A softmax layer based on decoder's last layer $s_{t,n}$ is used to gain a probability distribution $P_{predict}$ over target-side vocabulary.

$$p(y_t|y_1, ..., y_{t-1}, \mathbf{x}) = \text{softmax}(s_{t,n} * \mathbf{W}), \quad (1)$$

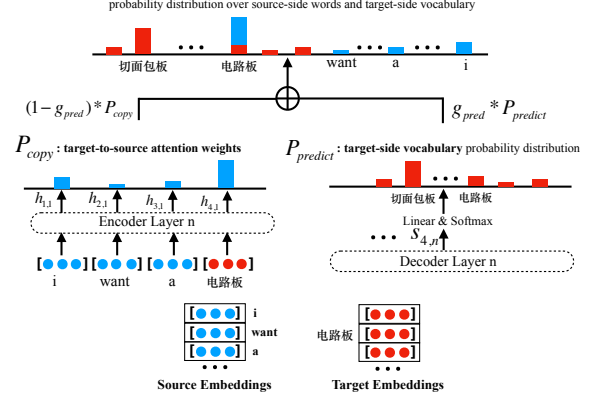where $\mathbf{W}$ is the weight matrix which is learned, $\mathbf{x}$

Figure 2: Shared embeddings and pointer network

represent the source sentence, $\{y_1, y_2, ..., y_t\}$ represent target words.

4.1 Shared Target Embeddings

Shared target embeddings enforces the correspondence between source-side and target-side expressions on the embedding level. As shown in Figure 2, during encoding, source-side target word embeddings are identical to their embeddings in the target-side vocabulary embedding matrix. This makes it easier for the model to copy source-side target words to the output.

4.2 Pointer Network

To strengthen copying through locating source-side target words, we integrate *pointer network* (Gulcehre et al., 2016) into the decoder, as shown in Figure 2. At each decoding time step t, the target-to-source attention weights $\alpha_{t,1}, ..., \alpha_{t,m}$ are utilized as a probability distribution P_{copy}, which models the probability of *copying* a word from the i-th source-side position. The i-th source-side position may represent a source-side word or a source-side target word. P_{copy} is added to $P_{predict}$, the probability distribution over target-side vocabulary, to gain a new distribution over both the source and the target side vocabulary[4]:

$$P = (1 - g_{pred}) * P_{copy} + g_{pred} * P_{predict}, \quad (2)$$

where g_{pred} is used to control the contribution of two probability distributions. For time step t, g_{pred} is calculated from the context vector c_t and the current hidden state of the decoder's last layer $s_{t,n}$:

---

[4]For the words which belong to the source-side vocabulary but are not appeared in the source-side sentence, the probabilities are set to 0.

$$g_{pred} = \sigma(c_t * W_p + s_{t,n} * W_q + b_r), \quad (3)$$

where W_p, W_q, and b_r are parameters trained and σ is the sigmoid function. In addition, the context vector c_t is calculated as $c_t = \sum_{i=1}^{m} \alpha_{t,i} * h_{i,n}$, where $\alpha_{t,i}$ is attention weight mentioned earlier. $\{h_{1,n}, h_{2,n}, ..., h_{m,n}\}$ are the source-side hidden states of the encoder's last layer.

5 Experiments

We compare our method with strong baselines on large-scale En-Ru and Ch-En tasks on various test sets across different domains, using a strongly optimized Transformer (Vaswani et al., 2017). BLEU (Papineni et al., 2002) is used for evaluation.

5.1 Data

Our training corpora are taken from the WMT2018 news translation task.

En-Ru. We use 13.88M sentences as baseline training data, containing both a real bilingual corpus and a synthetic back-translation corpus (Sennrich et al., 2015a). The synthetic corpus is translated from "NewsCommonCrawl", which can be obtained from the WMT task. The news domain contains four different test sets published by WMT2018 over the recent years, namely "news2015", "news2016", "news2017", and "news2018", respectively, each having one reference. The e-commerce domain contains four files totalling 7169 sentences, namely "subject17", "desc17", "subject18", and "desc18", respectively, each having one reference. The sentences are extracted from e-commerce websites, in which "subject"s are the goods names shown on a listing page. "desc"s refer to information in a commodity's description page. "subject17" and "desc17" are released[5]. Our development set is "news2015".

Ch-En. We use 7.42M sentences as our baseline training data, containing both real bilingual corpus and synthetic back-translation corpus (Sennrich et al., 2015a). We use seven public development and test data sets, four in the news domain, namely "NIST02", "NIST03", "NIST04", "NIST05", respectively, each with four references, and three in the spoken language domain, namely

---

[5] https://github.com/batman2013/e-commerce_test_sets

"CSTAR03", "IWSLT2004", "IWLST2005", respectively, each with 16 references. "NIST03" is used for development.

5.2 Experimental Settings

We use six self-attention layers for both the encoder and the decoder. The embedding size and the hidden size are set to 512. Eight heads are used for self-attention. A feed-forward layer with 2048 cells and Swish (Ramachandran et al., 2018) is used as the activation function. Adam (Kingma and Ba, 2014) is used for training; warmup step is 16000; the learning rate is 0.0003. We use label smoothing (Junczys-Dowmunt et al., 2016) with a confidence score of 0.9, and all the drop-out (Gal and Ghahramani, 2016) probabilities are set to 0.1.

We extract a SMT phrase table on the bilingual training corpus by using moses (Koehn et al., 2007) with default setting, which is used for matching sentence pairs to generate augmented training data. We apply count-based pruning (Zens et al., 2012) to the phrase table, the threshold is set to 10.

During **decoding**, similar to Hasler et al. (2018), Alkhouli et al. (2018) and Post and Vilar (2018), we make use of references to obtain gold constraints. Following previous work, pre-specified translations for each source sentence are sampled from references and used by all systems for fair comparison.

In all the baseline systems, the vocabulary size is set to 50K on both sides. For "Data augmentation", to allow the source-side dictionary to cover target-side words, the target- and source-side vocabularies are merged for a new source vocabulary. For "Shared embeddings", the source vocabulary remains the same as the baselines, where the source-side target words use embeddings from target-side vocabulary.

5.3 System Configurations

We use an in-house reimplementation of Transformer, similar to Google's Tensor2Tensor. For the baselines, we reimplement Crego et al. (2016), as well as Post and Vilar (2018). BPE (Sennrich et al., 2015b) is used for all experiments, the operation is set to 50K. Our test sets cover news and e-commerce domains on En-Ru, and news and spoken language domains on Ch-En.

Baseline 1: Using Placeholder. We combine Luong et al. (2014) and Crego et al. (2016). For

	news15	news16	news17	news18	△	subject17	desc17	subject18	desc18	△
Marian	33.27	31.91	36.18	32.11	-0.15	8.03	23.21	11.02	27.94	-0.46
Transformer	33.29	31.95	36.57	32.27	-	8.56	23.53	11.95	27.90	-
+ Placeholder	33.14	32.07	36.24	32.03	-0.15	9.81	24.04	13.84	29.34	+1.27
+ Lexi. Cons.	33.50	32.62	36.65	32.88	+0.39	9.24	23.67	13.1	29.83	+0.98
Data Aug.	34.71	33.69	38.43	33.51	+1.57	10.63	25.56	14.26	30.92	+2.36
+ Share	35.28	34.37	39.02	34.44	+2.26	10.82	25.84	15.20	30.97	+2.72
+ Share&Point	**36.44**	**35.31**	**40.23**	**35.43**	**+3.33**	**11.58**	**26.53**	**16.08**	**32.17**	**+3.61**

Table 1: Results on En-Ru, one or two source phrases of each sentence have pre-specified translation. "Transformer" is our in-house vanilla Transformer baseline. "Marian" is the implementation of Transformer by Junczys-Dowmunt et al. (2018), which is used as a reference of our Transformer implementation.

	CSTAR03	IWSLT04	IWSLT05	△	NIST02	NIST03	NIST04	NIST05	△
Transformer	53.03	56.52	64.72	-	40.52	37.85	40.12	39.26	-
+ Placeholder	52.51	56.15	64.44	-0.39	40.01	37.16	39.96	38.87	-0.44
+ Lexi. Cons.	53.30	56.95	65.63	+0.54	40.36	38.02	40.44	39.72	+0.20
Data Aug.	53.82	57.28	65.54	+0.79	40.85	38.41	40.81	40.29	+0.65
+Share	**53.90**	**57.67**	65.59	**+0.96**	41.06	38.57	41.22	40.38	+0.87
+Share&Point	53.79	57.29	**65.65**	+0.82	**41.11**	**38.7**	**41.3**	**40.4**	**+0.94**

Table 2: Results on Ch-En, one or two source phrases of each sentence have pre-specified translation.

generating placeholder tags during training, following Crego et al. (2016), we use a named entity translation dictionary which is extracted from Wikidata[6]. The dictionary is released together with e-commerce test sets, which is mentioned before. For Ch-En, the dictionary contains 285K person names, 746K location names and 1.6K organization names. For En-Ru, the dictionary contains 471K person names, 254K location names and 1.5K organization names. Additionally, we manually corrected a dictionary which contains 142K brand names and product names translation for En-Ru. By further leveraging word alignment in the same way as Luong et al. (2014), the placeholder tags are annotated with indices. We use FastAlign (Dyer et al., 2013) to generate word alignment. The amount of sentences containing placeholder tags is controlled to a ratio of 5% of the corpus. During decoding, pre-specified translations described in Section 5.2 are used.

Baseline 2: Lexical Constraints. We re-implement Post and Vilar (2018), integrating their algorithm into our Transformer. Target-side words or phrases of pre-specified translations mentioned in Section 5.2 are used as lexical constraints.

Our System. During training, we use the method described in Section 3.1 to obtain the augmented training data. The SMT phrase table mentioned in Section 5.2 is used for "Indexing" and "Sampling". During decoding, pre-specified translations mentioned in Section 5.2 are used. The augmented data contain sampled sentences with one or two replacements on the source side. By applying the two sampling steps described in Section 3.1, about 10M and 6M augmented Ch-En and En-Ru sentences are generated, respectively. The final training corpora consists of both the augmented training data and the original training data.

5.4 Results

Comparison with Baselines. Our Transformer implementation can give comparable performance with state-of-the-art NMT (Junczys-Dowmunt et al., 2018), see "Transformer" and "Marian" in Table 1, which also shows a comparison of different methods on En-Ru. The *lexical constraint* method gives improvements on both the news and the e-commerce domains, compared with the Transformer baseline. The *placeholder* method also gives an improvement on the e-commerce

Pre-Specified Translation: 计划生育 —> **planned parenthood**

Source: 上 个 月 ， 民主 党 否决 了 共和 党 提出 的 一 项 提议 ， 该 提议 要求 拨款 11 亿 美元 抗击 寨 卡 但 包 含 禁止 资助 计划 生育 协会 提供 与 病毒 相关 的 避孕 条件 的 条款 ， 而 寨 卡 病毒 可以 通过 性 途径 传播 。

Transformer: last month , the democratic party of the republic rejected a proposal by the republican party to grant $ 1.1 billion to fight against the zika but contained provisions prohibiting the **family planning** association from providing virus - related contraceptive conditions and the zika virus could be transmitted sexually .

Baseline1(Placeholder): last month , the party rejected a proposal from the republican party that would require $ 1.1 billion to fight zika , but would include provisions that prohibit **program** **planned parenthood** from providing virus - related contraceptive conditions , and zika could be transmitted sexually .

Baseline2(Lexi. Cons.): last month , the democratic party of the republic rejected a proposal by the republican party to grant $ 1.1 billion to fight against the zika but contained provisions that prohibit the family planning association from providing contraceptive conditions related to the virus and the zika virus could be transmitted through a sexual route **. the u . s . national security council has been banned from funding for the** **planned parenthood** .

Ours: last month , the democratic party rejected a republican proposal calling for $ 1.1 billion to fight zika , which contains provisions that prohibit the association of **planned parenthood** from providing virus - related contraceptive conditions , and the zika virus could be transmitted sexually .

Reference: last month , democrats blocked consideration of a republican measure that would have allocated $ 1.1 billion to fight zika but included provisions that would have banned funding for **planned parenthood** to provide contraception related to the virus , which can be sexually transmitted .

Figure 3: Sample outputs.

domain. The average improvement is calculated over all the test set results in each domain. In the news domain, the average improvement of *our method* is 3.48 BLEU higher compared with *placeholder*, and 2.94 over *lexical constraints*. In the e-commerce domain, the average improvement of *our method* is 1.34 BLEU compared with *placeholder*, and 2.63 with *lexical constraints*. Both shared embedding and pointer network are effective. Table 2 shows the same comparison on Ch-En. In the spoken language domain, the average improvement is 1.35 BLEU compared with *placeholder*, and 0.42 with *lexical constraints*. In the news domain, the average improvement is 1.38 BLEU compared with *placeholder*, and 0.74 with *lexical constraints*.

We find that the *placeholder* method can only bring improvements on the En-Ru e-commerce test sets, since the pre-specified translations of the four e-commerce test sets are mostly *entities*, such as brand names or product names. Using placeholder tags to represent these entities leads to relatively little loss of word meaning. But on many of the other test sets, pre-specified translations are mostly vocabulary words. The placeholder tags fail to keep their word meaning during translation, leading to lower results.

The speed contrast between unconstrained NMT, lexical constraint and our method is shown in Table 3. The decoding speed of our method is equal to unconstrained NMT, and faster than the lexical constraint method, which confirms our in-

Beam Size	5	10	20	30
Unconstrained & Ours	416	312	199	146
Lexical Constraint	102	108	74	50

Table 3: Decoding speed (words/sec), Ch-En dev set.

tuition introduced earlier.

Sample Outputs. Figure 3 gives a comparison of different system's translations. Given a Chinese source sentence, the baseline system fails to translate "计划生育" adequately, as "family planning" is not a correct translation of "计划生育". In the pre-specified methods, the correct translation ("计划生育" to "planned parenthood") is achieved through different ways.

For the *placeholder* method, the source phrase "计划生育" is replaced with the placeholder tag "tag_1" during pre-processing. After translation, output "tag_1" is replaced with "planned parenthood" as a post-processing step. However, the underlined word "program" is generated before "planned parenthood", which has no relationship with any source-side word. The source-side word "协会", which means "association", is omitted in translation. Through deeper analysis, the specific phrase "program tag_1" occurs frequently in the training data. During decoding, using the hard tag leads to the loss of the source phrase's original meaning. As a result, the word "program" is incorrectly generated along with "tag_1".

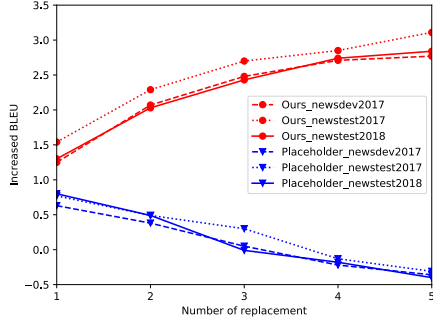The *lexical constraints* method regards the tar-
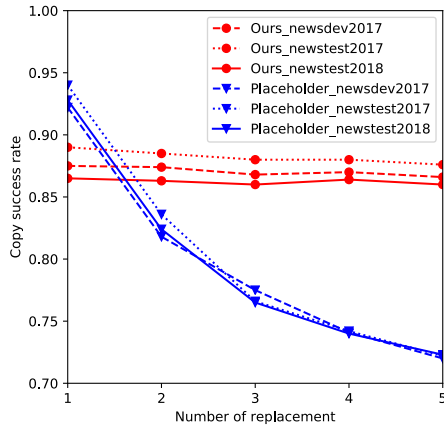
Figure 4: Increased BLEU on Ch-En test sets.

Figure 5: Copy success rate on Ch-En test sets.

get side of the pre-specified translation as a lexical constraint. Here the altered beam search algorithm fails to predict the constraint "planned parenthood" during previous decoding steps. Although the constraint finally comes into effect, over translation occurs, which is highlighted by the underlined words. This is because the method enforces hard constraints, preventing decoding to stop until all constraints are met.

Our method makes use of pre-specified translation by replacing the source-side phrase "计划生育" with the target-side translation "planned parenthood", copying the desired phrase to the output along with the decoding procedure. The translation "association of planned parenthood from providing" is the exact translation of the source-side phrase "计划(planned) 生育(parenthood) 协会(association) 提供(providing)", and agrees with the reference, "planned parenthood to provide".

5.5 Analysis

Effect of Using More Pre-specified Translations. Even though the augmented training data have only one or two replacements on the source side, the model can translate a source sentence with up to five replacements. Figure 4 shows that compared with unconstrained Transformer, the translation quality of our method keeps increasing when the number of replacements increases, since more pre-specified translations are used.

We additionally measure the effect on the Ch-En WMT test sets, namely "newsdev2017", "newstest2017", "newstest2018", respectively, each having only one reference instead of four. The baseline BLEU scores on these three test sets are 18.49, 20.01 and 19.05, respectively. Our method gives BLEU scores of 20.56, 22.3, 21.08, respectively, when using one or two pre-specified translations for each sentence. The increased BLEU when utilizing different number of pre-specified translations is shown in Figure 4. We found that the improvements on WMT test sets are more significant than on NIST, since pre-specified translations are sampled from one reference only, enforcing the output to match this reference. The placeholder method does not give consistent improvements on news test sets, due to the same reason as mentioned earlier.

As shown in Figure 5, the copy success rate of our method does not decrease significantly when the number of replacements grows. Here, a *copy success* refers a pre-specified target translation that can occur in the output. The placeholder method achieves a higher copy success rate than ours when the number of replacements is 1, but the copy success rate decreases when using more pre-specified translations. The copy success rate of the lexical constraint method is always 100%, since it imposes hard constraints rather than soft constraints. However, as discussed earlier, overall translation quality can be harmed as a cost of satisfying decoding constraints by their method.

In the presented experiment results, the highest copy success rate of our method is 90.54%, which means a number of source-side target words or phrases are not successfully copied to the translation output. This may be caused by the lack of training samples for certain target-side words or phrases. In En-Ru, we additionally train a model with augmented data that is obtained by matching

	NIST02	NIST03	NIST04	NIST05
Data Aug.	83.89%	85.71%	86.71%	87.45%
+Share&Point	87.72%	88.31%	89.18%	90.54%

Table 4: Copy success rate on Ch-En test sets.

	news15	news16	news17	news18
Baseline	33.29	31.95	36.57	32.27
Ours	33.53	32.29	36.54	32.47

Table 5: BLEU scores of non code-switched (original) input on En-Ru test sets.

an SMT phrase table without any pruning strategy. The copy success rate can reach 98%, even without using "shared embedding" and "pointer network" methods.

Effect of Shared Embeddings and Pointer Network. The gains of *shared embeddings* and *pointer network* are reflected in both the copy success rate and translation quality. As shown in Table 4, when using one pre-specified translation for each source sentence, the copy success rate improves on various test sets by integrating shared embeddings and pointer network, demonstrating that more pre-specified translations come into effect. Table 1 and Table 2 earlier show the improvement of translation quality.

Translating non Code-Switched Sentences. Our method preserves its strength on translating non code-switched sentences. As shown in Table 5, the model trained on the augmented corpus has comparable strength on translating unreplaced sentences as the model trained on the original corpus. In addition, on some test sets, our method is slightly better than the baseline when translating non code-switched source sentences. This can be explained from two aspects: First, the augmented data make the model more robust to perturbed inputs; Second, the pointer network makes the model better by copying certain source-side words (Gulcehre et al., 2016), such as non-transliterated named entities.

6 Conclusion

We investigated a data augmentation method for constraining NMT with pre-specified translations, utilizing code-switched source sentences and their translations as augmented training data. Our method allows the model to learn to translate

source-side target phrases by "copying" them to the output, achieving consistent improvements over previous lexical constraint methods on large NMT test sets. To the best of our knowledge, we are the first to leverage code switching for NMT with pre-specified translations.

7 Future Work

In the future, we will study how the copy success rate and the BLEU scores interact when different sampling strategies are taken to obtain augmented training corpus and when the amount of augmented data grows. Another direction is to validate the performance when applying this approach to language pairs that contain a number of identical letters in their alphabets, such as English to French and English to Italian.

Acknowledgments

References

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. *arXiv preprint arXiv:1809.03985*.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Guided open vocabulary image captioning with constrained beam search. *CoRR*, abs/1612.00576.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. *arXiv preprint arXiv:1606.02006*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

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.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard,

Yongchao Deng, et al. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. *arXiv preprint arXiv:1603.08148*.

Eva Hasler, Adrià De Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. *arXiv preprint arXiv:1805.03750*.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Rico Sennrich. 2016. The amu-uedin submission to the wmt16 news translation task: Attention-based nmt models as feature functions in phrase-based smt. *arXiv preprint arXiv:1605.04809*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2016. Neural name translation improves neural machine translation. *arXiv preprint arXiv:1607.01856*.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318, Philadelphia, Pennsylvania, USA.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *arXiv preprint arXiv:1804.06609*.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2018. Searching for activation functions.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *Computer Science*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. 2017a. Neural machine translation advised by statistical machine translation. In *AAAI*, pages 3330–3336.

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.

Richard Zens, Daisy Stanton, and Peng Xu. 2012. A systematic comparison of phrase table pruning techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 972–983. Association for Computational Linguistics.