Controlling Target Features in Neural Machine Translation via Prefix Constraints

Shunsuke Takeno[†*] **Masaaki Nagata**[‡] **Kazuhide Yamamoto**[†]
[†]Nagaoka University of Technology,
1603-1 Kamitomioka, Nagaoka, Niigata, 940-2188 Japan
{takeno, yamamoto}@jnlp.org
[‡]NTT Communication Science Laboratories, NTT Corporation,
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
nagata.masaaki@labs.ntt.co.jp

Abstract

We propose *prefix constraints*, a novel method to enforce constraints on target sentences in neural machine translation. It places a sequence of special tokens at the beginning of target sentence (target prefix), while side constraints (Sennrich et al., 2016) places a special token at the end of source sentence (source suffix). Prefix constraints can be predicted from source sentence jointly with target sentence, while side constraints must be provided by the user or predicted by some other methods. In both methods, special tokens are designed to encode arbitrary features on target-side or metatextual information. We show that prefix constraints are more flexible than side constraints and can be used to control the behavior of neural machine translation, in terms of output length, bidirectional decoding, domain adaptation, and unaligned target word generation.

1 Introduction

It is difficult to change the behaviors of a current neural machine translation system, because the internal states of the system are represented by vectors of real numbers. There are no symbols to be manipulated and end-to-end optimization makes it impossible to identify the source of poor performance.

Some studies control the output of the encoder-decoder model, through the use of additional information such as target-side information and meta-textual information. Target-side information includes politeness (Sennrich et al., 2016), voice (Yamagishi et al., 2016), sentence length (Kikuchi et al., 2016), and target language (Johnson et al., 2016). Meta-textual information include dialogue act (Wen et al., 2015), user personality (Li et al., 2016), topic (Chen et al., 2016), and domain (Kobus et al., 2016)

Two approaches can be used to provide additional information to the encoder-decoder model, word-level methods and sentence-level methods. Word-level methods encode the additional information as a vector (embedding) that is input together with a word at each time step of either (or both) encoder and decoder (Wen et al., 2015; Li et al., 2016; Kikuchi et al., 2016). Sentence level methods encode the additional information as special tokens. Side constraints are placed at the end of source sentence (Sennrich et al., 2016; Johnson et al., 2016; Yamagishi et al., 2016), while our proposal, prefix constraints, is placed at the beginning of target sentence.

The advantage of sentence-level methods over word-level methods is their simplicity in application. The network structure of the underlying encoder-decoder model does not have to be modified. The problem with side constraints is that, at test time, additional information must be either specified by the user or automatically predicted by some other method. As prefix constraints move the special tokens from source to target, they can be predicted by the network jointly with target sentence. Like side constraints, the user can specify prefix constraints by using prefix-constrained decoding (Wuebker et al., 2016), which can be implemented by a trivial modification of the decoder.

The following sections start by describing the framework of prefix constraints. We then show three simple use cases, namely, length control, bidirectional decoding, and domain adaptation. We then show a more sophisticated usage of prefix constraints: unaligned target word generation.

[*]Currently, Retty Inc.

Proceedings of the 4th Workshop on Asian Translation, pages 55–63,
Taipei, Taiwan, November 27, 2017.

2 Encoder-Decoder Model with Prefix Constraints

2.1 Encoder-Decoder Model

First, we briefly describe the attention-based encoder-decoder model (Bahdanau et al., 2015; Luong et al., 2015), which is the state-of-the-art neural machine translation method and the baseline of this study.

Given input sequence $\boldsymbol{x} = x_1 \dots x_n$ and model parameters θ, the encoder-decoder model formulates the likelihood of the output sequences $\boldsymbol{y} = y_1 \dots y_m$ as follows:

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{j=1}^{m} \log p\left(y_j|\boldsymbol{y_{<j}}, \boldsymbol{x}; \theta\right) \quad (1)$$

The encoder is a recurrent neural network (RNN) which projects input sequence $\boldsymbol{x}$ into a sequence of hidden states $\boldsymbol{h} = h_1 \dots h_n$ via non-linear transformation. The decoder is another RNN which predicts target words $\boldsymbol{y}$ sequentially, one word at a time. The encoder-decoder model is trained to maximize the conditional likelihood on a parallel corpus by stochastic gradient descent.

$$J = -\sum_{(x,y)\in D} \log p(\boldsymbol{y}|\boldsymbol{x}) \quad (2)$$

Attention-based encoder-decoder models have an additional single-layer feed-forward neural network, called attention layer. It calculates a weight for each source word x_i to predict target word y_j from previous target word y_{j-1} and hidden states of the encoder h_i.

2.2 Side Constraints

Sennrich et al. (2016) proposed a method to control the level of politeness in target sentence in English-to-German translation. They add a T-V distinction tag at the end of the source sentence, so that target sentence is either familiar (Latin *Tu*) or polite (Latin *Vos*).

Are you kidding? [T] → Machst du Witze?
Are you kidding? [V] → Machen Sie Witze?

In their method, the features that the generated target sentence must satisfy are called *side constraints*. At training time, the correct feature is extracted from the sentence pair. At test time, the special token is assumed to be provided by the user. Automatic prediction of the side constraints from the source sentence at test time is an open problem. Johnson et al. (2016) used the framework for multilingual translation.

2.3 Prefix Constraints

In our proposed method, a sequence of special tokens is placed at the beginning of the target sentence. In other words, they are the prefix to the extended target sentence.

Let a sequence of features extracted from a pair of source sentence $\boldsymbol{x}$ and target sentence $\boldsymbol{y}$ be $\boldsymbol{c} = c_1 \dots c_k$, and extended target sentence be $\tilde{\boldsymbol{y}} = \boldsymbol{cy}$. The baseline encoder-decoder model Eq. (1) is extended as follows.

$$\log p(\tilde{\boldsymbol{y}}|\boldsymbol{x}) = \log p(\boldsymbol{c}|\boldsymbol{x}) + \log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{c}) \quad (3)$$

$$\log p(\boldsymbol{c}|\boldsymbol{x}) = \sum_{j=1}^{k} \log p\left(c_j|\boldsymbol{c_{<j}}, \boldsymbol{x}; \theta\right)$$

$$\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{c}) = \sum_{j=1}^{m} \log p\left(y_j|\boldsymbol{y_{<j}}, \boldsymbol{x}, \boldsymbol{c}; \theta\right)$$

and the objective function becomes

$$J = -\sum_{(x,y)\in D} \log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{c}) + \log p(\boldsymbol{c}|\boldsymbol{x}). \quad (4)$$

Prefix constraints can be either automatically predicted or specified by the user. In the default use of the decoder, both prefix and target sentence are jointly generated (predicted) from source sentence. Prefix can be specified by using *prefix-constrained decoding* (Wuebker et al., 2016), which is a beam search method that constrains the output to match a specified prefix. In the constrained mode, we feed c_j directly to the next time step regardless of the current prediction of the decoder. Once the specified prefix has been utilized, the decoder switches to standard (unconstrained) beam search and the most probable word y_j is passed to the next time step.

3 Basic Examples

3.1 Length Control

The first example encodes the desired length of the target sentence for length control.

京都が好きです → #3 I love Kyoto

Length control is extremely be useful when translating headlines, captions, and subtitles. Kikuchi et al. (2016) proposed four methods to control the length of the sentence generated by an encoder-decoder model in a text summarization task; they considered two learning-based methods using length embeddings, namely *LenEmb* and *LenInit*. *LenEmb* method explicitly enters the remaining length to the decoder at each time step,

while *LenInit* method enters the desired length once at the initial state of the decoder. They designed a dedicated network structure for each method.

In spirit, our method is similar to the *LenInit* method, but we don't have to modify the underlying network structure. Note that we do not tell the network that '#3' is the length of the target sentence. The network automatically learns the meaning of the symbol from the regularity of the training data and then calculates its embedding.

3.2 Bidirectional Decoding

The second example encodes the decoding direction of target sentence for bidirectional decoding.

京都が好きです → #L2R I love Kyoto
京都が好きです → #R2L Kyoto love I

We make two sentence pairs, one with prefix '#L2R' (left-to-right) and the other with prefix '#R2L' (right-to-left) for each sentence pair, and train a single model. At test time, given an input sentence, the decoder automatically selects optimal decoding direction either '#L2R' or '#R2L' depending on their probabilities.

Liu et al. (2016) proposed a target-bidirectional decoding method that encourages the agreement between left-to-right and right-to-left decoding. Their method requires two separate models, one for left-to-right n-best decoding and the other for right-to-left rescoring, and an additional mechanism for encouraging agreement (rescoring). Our method implements bidirectional decoding in one pass decoding without changing the underlying network structure.

3.3 Domain Adaptation

The third example encodes dataset names of a bilingual text for domain adaptation. Here, IWSLT is a travel expression corpus and KFTT is a corpus of Japanese Wikipedia pages on Kyoto and its English translation.

朝食はいくらですか。
→ #IWSLT How much is the breakfast ?
妙法蓮華経を根本経典とする。
→ #KFTT Its fundamental sutra is lotus sutra .

Kobus et al. (2016) proposed a domain adaptation method using side constraints. They used a separate classifier for predicting the domain of a sentence before translation if it is not known. Li et al. (2016) used Speaker IDs of Twitter to add personality to a conversational agent. Speaker embeddings are learned jointly with word embeddings and entered into the decoder at each time step. Luong and Manning (2015) proposed a domain adaptation method based on fine tuning in which an out-of-domain model is further trained on in-domain data.

Our method can automatically predict domain jointly with target sentence. We don't have to change the underlying network structure and domain embeddings are jointly learned with word embeddings as a part of target vocabulary. One of the potential benefits of our method is that only one model is made and used for all domains. If multiple domains must be supported, the methods based on fine tuning (Luong and Manning, 2015) have to make a model for each domain.

4 Unaligned Target Word Generation

The fourth example encodes information on unaligned target words for generating target sentences. It is significantly more complex than the previous examples. We first describe its motivation and then derive two types of prefix constraints.

4.1 Unaligned Target Words

Given a pair of sentences that are translations of each other, some words in one language cannot be aligned to any words in the other language. We call them *unaligned words*.

Japanese case markers such as が *(ga)*, を *(wo)*, に *(ni)* and English articles such as *a*, *an*, *the* do not have counterparts in the other language. Other than these grammatical differences between two languages, unaligned words can be caused by specific linguistic phenomena in one language, such as zero pronouns (dropped subject and object) in Japanese and expletives in English (*there* in there-construction, *do* in interrogative sentence, *it* in formal subject, etc.).

In machine translation, unaligned words in target sentence are problematic because the information required for translation is not explicitly present in the source sentence. There are many works that aim at improving machine translation performance by supplementing unaligned words, but they focus on specific linguistic phenomena such as Japanese case marker (Hisami and Suzuki, 2007), Chinese zero pronoun (empty category) (Chung and Gildea, 2010;

Xiang et al., 2013; Wang et al., 2016), Japanese zero pronoun (Taira et al., 2012; Kudo et al., 2014) and English determiner (Tsvetkov et al., 2013). There are no language independent methods that can cope with unaligned target words.

4.2 Identifying Unaligned Target Words

We first propose a language independent method for automatically identifying unaligned target words. We assume word alignment is given for a bilingual sentence pair, where NULL represents empty word. We define a score $S_u(w)$. It represents the likelihood that a word w in target sentence e aligns to the NULL in source sentence f. The most straightforward way to define $S_u(w)$ is to use the word translation probability obtained from the word alignment in the training corpus,

$$S_u(w) = p(f = NULL|e = w) \quad (5)$$

Our preliminary experiment showed that the scores yielded by Eq. (5) are not reliable for low frequency target words. We therefore use the following equation to filter out low frequency NULL-generated target words.

$$S_u(w) = p(e = w|f = NULL) * p(f = NULL|e = w) \quad (6)$$

We use GIZA++ (Och and Ney, 2003) to obtain word alignment for both translation directions. Word alignment is symmetrized by *intersection* heuristics, because the word alignment obtained by *grow-diag-final-and*, is noisy for unaligned words.

Table 1 shows the top 50 unaligned target words as determined by Eq. (6) in the IWSLT-2005 Japanese-to-English translation dataset, which is described in the experiment section. We can see that the automatically extracted unaligned target words include zero pronouns (*i, you, it*), articles (*a, the*), light verbs (*take, get, make*), and expletives (*do, does*).

4.3 Prefix Constraints for Unaligned Target Words

We propose here two types of prefix constraints for improving the translation of unaligned target words: LEX and COUNT.

LEX places a sequence of unaligned target words at the beginning of the target sentence in the same order they appear in the target sentence. A special token, #GO, is added to delimit the variable length prefix relative to target sentence. In the following examples, words with underline indicate unaligned target words.

赤ワインを頂けますか。
→#i #GO may i have some red wine ?
では当日御待ちして居ります。
→#we #you #GO we are waiting for you that day .

COUNT uses the number of unaligned target words as a prefix. As shown in the following examples, the number of unaligned target words are surrounded by "[" and "]" to distinguish the (fixed length) prefix from target sentence [1].

赤ワインを頂けますか。
→ [1] may i have some red wine ?
では当日御待ちして居ります。
→ [2] we are waiting for you that day .

5 Experiment

5.1 Datasets and Tools

The experiments used five publicly available Japanese-English parallel corpora, namely IWSLT-2005, KFTT, GVOICES, REUTERS, and TATOEBA, as shown in Table 2. IWSLT-2005 is a dataset for Japanese-English Tasks of the International Workshop on Spoken Language Translation (Eck and Hori, 2005). It is available from ALAGIN[2]. KFTT (Kyoto Free Translation Task) is a Japanese-English translation task on Wikipedia articles related to Kyoto[3]. Parallel Global Voices is a multilingual corpus created from Global Voices websites which translate social media and blogs (Prokopidis et al., 2016). Tatoeba is a collection of multilingual translated example sentences from Tatoeba website. These last two are available from OPUS (Tiedemann, 2012). Reuters are Japanese-English parallel corpus made by aligning Reuters RCV1 RCV2 multilingual text categorization test collection data set (RCV1 for English and RCV2 for other languages) available from NIST (Utiyama and Isahara, 2003)[4].

The unaligned target word generation experiments used two additional proprietary spoken

[1]The COUNT feature can be thought of a substitute for the fertility of the IBM model (Brown et al., 1993), or the generative model for NULL-generated target words (Schulz et al., 2016).

[2]http://alagin.jp/

[3]http://www.phontron.com/kftt/index.html

[4]The aligned parallel corpus is available from the homepage of the first author of (Utiyama and Isahara, 2003)

	$S_u(w)$		$S_u(w)$		$S_u(w)$		$S_u(w)$		$S_u(w)$
i	0.263	like	0.119	's	0.090	be	0.070	want	0.060
the	0.233	of	0.118	can	0.090	'll	0.070	that	0.059
a	0.214	me	0.114	'm	0.089	take	0.068	there	0.059
you	0.171	in	0.109	at	0.084	would	0.068	one	0.054
,	0.166	my	0.108	how	0.082	and	0.067	could	0.051
it	0.155	have	0.101	some	0.078	what	0.067	was	0.051
to	0.133	on	0.098	your	0.077	get	0.066	make	0.051
for	0.132	we	0.097	will	0.075	any	0.066	this	0.049
do	0.129	'd	0.094	with	0.074	an	0.064	here	0.049
please	0.126	is	0.091	are	0.074	does	0.063	by	0.048

Table 1: Top 50 unaligned target words in IWSLT2005

Name	Label	Sents.	len.(ja)	len.(en)
IWSLT-2005	train	19,972	9.9	9.4
(Conversation)	dev	506	8.1	7.5
	test	1,000	8.2	7.6
KFTT	train	440,288	27.0	26.3
(Wikipedia)	dev	1,235	27.8	25.1
	test	1,160	24.5	23.5
GVOICES	train	43,488	26.3	19.8
(Blog)	dev	1,000	25.1	18.9
	test	1,000	28.7	21.2
REUTERS	train	54,011	34.3	25.2
(News)	dev	1,000	34.4	25.2
	test	1,000	34.6	25.5
TATOEBA	train	185,426	10.1	9.14
(Examples)	dev	1,000	10.2	9.21
	test	1,000	11.8	9.23
ALL	train	753,185	23.3	21.2
	dev	4,741	23.2	18.6
	test	5,160	22.2	17.5

Table 2: Datasets Statistics

language corpora as the IWSLT-2005 dataset is very small. One is the *Daionsen* parallel sentence database, made by Straightword Inc[5], which is a phrase book for daily conversation. It has 50,709 sentences with 431,258 words in English and 471,677 words in Japanese. The other is the HIT (Harbin Institute of Technology) parallel corpus (Yang et al., 2006) developed for speech translation. It is a collection of 62,727 sentences with 635,809 words in English and 796,200 words in Japanese. We call this dataset IWSLT-2005+EXTRA.

English sentences are tokenized and lowercased by the scripts used in Moses (Koehn et al., 2007). Japanese sentences are normalized by NFKC (a unicode normalization form) and word segmented by MeCab[6] with UniDic. For neural machine translation, we used seq2seq-attn[7], which implements an attention-based encoder-decoder (Luong et al., 2015). We used default settings unless otherwise specified. Translation accuracy is measured by BLEU (Papineni et al., 2002).

5.2 Length Control

Table 3 compares side constraints with prefix constraints in terms of length control for IWSLT-2005 dataset. Baseline is a NMT system trained on the parallel corpus without length tag. Side Constraints and Prefix Constraints stand for NMT systems trained on the corpus with length tags placed at the end of source sentence and at the begging of target sentence, respectively. In None, source sentences without length tag are entered into the system at test time. In Oracle, reference length is encoded as length tag and prefix constrained decoding is used in Prefix Constraints. In the training for Side Constraints, we mixed tagged sentences and non-tagged sentences to avoid over-fitting to length tag as described in (Sennrich et al., 2016).

	None	Oracle
Baseline	34.8	-
Side Constraints	33.0	35.4
Prefix Constraints	31.7	**35.7**

Table 3: Comparison between side constraints and prefix constraints on length control

As shown in Table 3, Prefix Constraints are comparable to or better than Side Constraints in controlling the length of the target sentence if the correct length is known and provided as an oracle. It is difficult to predict the length of target sentence from source sentence, which lowered the ac-

[5] http://www.straightword.jp/

[6] http://taku910.github.io/mecab/

[7] https://github.com/harvardnlp/seq2seq-attn

curacy of Prefix Constraints for None. The accuracy of length prediction for short sentences (less than 10 words) is 97.7%, while that for long sentences (more than or equal to 10 words) is 45.7%.

We found that length control for short sentence worked surprisingly well. The following is an example of prefix constrained decoding where length tags were changed from #2 to #9 for the source sentence どういたしまして (You're welcome). All of them are acceptable and have the specified length.

どういたしまして →
#2 anytime .
#3 you welcome .
#4 you 're welcome .
#5 you 're welcome up .
#6 you 're welcome , sir .
#7 you 're welcome . thank you .
#8 you 're welcome . you 're welcome .
#9 it 's all right . you 're welcome .

5.3 Bidirectional Decoding

	IWSLT	KFTT	REUTERS
L2R	34.8	20.9	19.7
R2L	32.8	20.1	19.6
Target-Bidi	**35.8**	21.1	20.2
Predict-Dir	35.6	**21.5**	**20.6**

Table 4: Comparison of target bidirectional method (Liu et al., 2016) and decoding direction prediction using prefix constraints

Table 4 shows a comparison between our implementation of target-bidirectional method (Target-Bidi) (Liu et al., 2016) and decoding direction prediction using prefix constraints (Predict-Dir) on IWSLT-2005, KFTT, and REUTERS datasets. L2R and R2L are baseline NMT system with left-to-right and right-to-left decoding, respectively. For the evaluation of Predict-Dir, sentences with '#R2L' tags are reversed and both '#L2R' and '#R2L' tags are removed. Predict-Dir is comparable to or better than Target-Bidi. Considering the simplicity of the proposed method, it is a viable option for bidirectional decoding.

5.4 Domain adaptation

Table 5 shows BLEU scores for the five datasets for different systems in terms of domain adaptation techniques. In Single, for each domain (dataset), the translation model is trained using only each dataset in isolation. In Join, one translation model is trained using a corpus made by simply concatenating all datasets without domain tags. In Predict and Oracle, one translation model is trained using a corpus made by concatenating all datasets with domain tags as target prefix. In Predict, domain tag is automatically predicted, while in Oracle, the domain tag of the reference is provided and used for prefix constrained decoding.

Comparing Single and Join, Small corpora such as GVOICES and REUTERS benefit most when additional parallel data is used, while the largest corpus KFTT experiences no such benefit. By adding domain tags (Predict and Oracle), all corpora including the largest KFTT can benefit from the combination of data sources. As the difference in accuracy between Predict and Oracle is small, we assume the domain prediction accuracy for the proposed method is high enough for the task.

In order to understand what is happening when domain tags are used as prefix constraints, we randomly selected 100 sentences from each dataset and calculated the hidden states for each reference. We then visualized the hidden state of the last layer of the decoder in the first time step (before domain tag entered) and the second time step (after domain tag entered) using t-SNE in Figure 1.

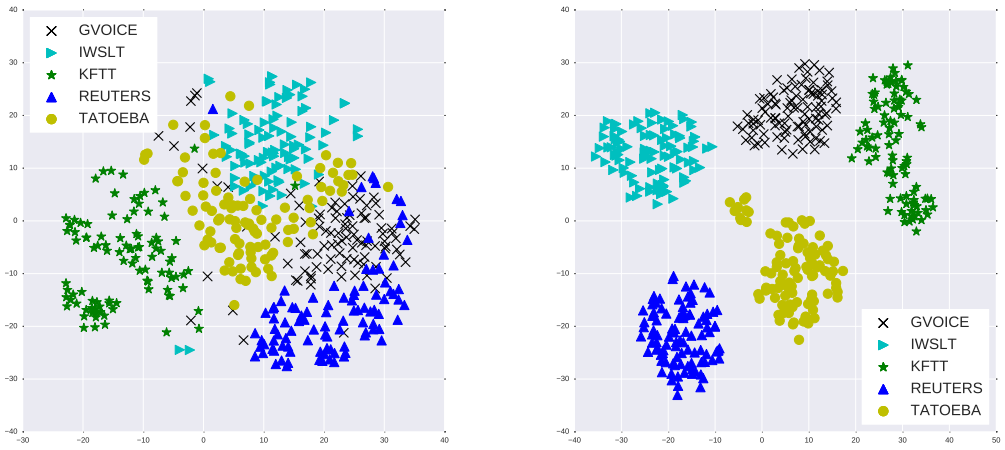
The figure shows the proximity between domains. In the initial step of the decoder, some domains such as IWSLT-2005 and TATOEBA, or GVOICES and REUTERS are very close each other. After domain tags are entered, all domains are clearly separated. Specifying the domain tag corresponds to moving the point in the figure from one cluster to another.

5.5 Unaligned target word generation

We made two lists of unaligned target words, top 10 and top 50, based on Eq. (6). For each sentence in the training data, unaligned target words were identified and used to make prefix constraints if they are in the list and unaligned in the sentence pair. Table 6 shows translation accuracy when COUNT and LEX are used as prefix constraints, where the candidates of target unaligned words are either top-10 or top-50. Baseline is the attention-based encoder-decoder model without prefix constraints. In Predict, prefix constraints are predicted from source sentence. In Oracle, prefix constraints are specified using reference target sentence and prefix constrained decoding is used.

	Single	Join	Predict	Oracle
GVOICES (43k sents.)	6.31	16.9	17.0	17.1
IWSLT (20k sents.)	34.8	36.8	37.1	37.1
KFTT (440k sents.)	20.9	20.8	21.1	21.1
REUTERS (54k sents.)	19.7	24.6	25.0	25.0
TATOEBA (185k sents.)	36.0	59.4	59.5	59.7

Table 5: BLEU scores for different systems in terms of domain adaptation techniques



(a) Before domain tag entry

(b) After domain tag entry

Figure 1: t-SNE visualization of the hidden states of the decoder for various domains

		IWSLT-2005+EXTRA	
	#UTW	Predict	Oracle
Baseline		36.5	-
COUNT	10	**37.5**	38.1
	50	36.9	38.0
LEX	10	36.4	41.7
	50	32.4	**46.9**

Table 6: Translation accuracy of prefix constraint prediction and prefix-constrained decoding

As for Predict, COUNT is significantly better than Baseline (about 1 BLEU point) when the small list of unaligned words, top-10, is used. It shows that translation accuracy can be improved by predicting prefix constraints and generating target sentence at the same time [8].

The accuracies for Oracle show that translation accuracy can be greatly improved if the user provides some information on unaligned target words.

[8]The average numbers of unaligned target words in train, dev, test set of IWSLT-2005+EXTRA are 3.1, 2.5, 2.6, respectively

	Precision	Recall
i	76	68
you	72	78
it	61	67

Table 7: Precision and recall of pronouns

If the number of unaligned words is provided, translation accuracy can be improved by about 3 BLEU points, and if the correct list of unaligned target words is provided, it can be improved by about 10 points. There is still much room for improvement as regards the problem of unaligned target words.

Table 7 shows precision and recall of unaligned target pronouns when COUNT based on top-10 list is used for prefix constraint prediction and the dataset is IWSLT-2005+EXTRA. We think the accuracies of around 70% are reasonable considering that some pronouns are context dependent.

Table 8 is a real example of the outputs of LEX and COUNT. In fact, it is very difficult to predict the correct set of unaligned words from just the

Input	いつ で も 話し合い に 応じる 準備 は できて いる から 、 ゴー サイン を 送って 下さい 。
Reference	#i #GO i 'm ready to start talks anytime so just say when .
Baseline	you 're always ready to talk to me , so you 'll have to have a thorough signature .
Predict (LEX)	#you #have #to #and #you #GO you 're ready to let us have ready and sent them to you .
Predict (COUNT)	[4] you 're always ready to talk with us . please send us a liqueur .

Table 8: A real example of the outputs of LEX and COUNT

source sentence without context. Leaving aside the errors caused by the unknown Japanese words ゴーサイン (go-ahead, green light, literally "go-sign"), the major challenge here is the Japanese zero subject. It could be "i", "you", "he/she", and depends on the context. In the other words, Oracle (LEX) is significantly better than Baseline because this kind of context dependent information is provided from the outside.

6 Conclusion

In this paper, we showed that prefix constraints can be used as a general framework for controlling the target features commonly needed in neural machine translation, such as length control, bidirectional decoding, domain adaptation, and unaligned target word generation.

There are many issues that must be tackled: For length control, translation accuracy could be improved if we can accurately predict the length of target sentence from source sentence. For domain adaptation, rigorous comparison between prefix constraints with other domain adaptation techniques, such as side constraints (Kobus et al., 2016), fine tuning (Luong and Manning, 2015), and their combination (Chu et al., 2017), are required to realize its full effectiveness. For unaligned target word generation, applying the proposed method to other domains such as news articles and other language pairs such as Chinese-to-English is required to show its generality.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the ICLR-2015*.

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

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided Alignment Training for Topic-Aware Neural Machine Translation. In *Proceedings of AMTA-2016*.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the ACL-2017*.

Tagyoung Chung and Daniel Gildea. 2010. Effects of Empty Categories on Machine Translation. In *Proceedings of the EMNLP-2010*. pages 636–645.

Matthias Eck and Chiori Hori. 2005. Overview of the iwslt 2005 evaluation campaign. In *Proceedings of the IWSLT-2005*. pages 1–22.

Toutanova Hisami and Kristina Suzuki. 2007. Generating case markers in machine translation. In *Proceedings of the NAACL-HLT-2007*. pages 49–56.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viegas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *arXiv preprint arXiv:1611.04558* .

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling Output Length in Neural Encoder-Decoders. In *Proceedings of the EMNLP-2016*. pages 1328–1338.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *arXiv preprint arXiv:1612.06140* .

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, and Ondrej Bojar and. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL-2007*.

Taku Kudo, Hiroshi Ichikawa, and Hideto Kazawa. 2014. A joint inference of deep case analysis and zero subject generation for Japanese-to-English statistical machine translation. In *Proceedings of the ACL-2014*. pages 557–562.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A Persona-Based Neural Conversation Model. In *Proceedings of the ACL-2016*. pages 994–1003.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the NAACL-HLT-2016*. pages 411–416.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the IWSLT-2015*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the EMNLP-2015*.

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

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

Prokopis Prokopidis, Vassilis Papavassiliou, and Stelios Piperidis. 2016. Parallel global voices: a collection of multilingual corpora with citizen media stories. In *Proceedings of the LREC-2016*.

Philip Schulz, Wilker Aziz, and Khalil Sima'an. 2016. Word alignment without null words. In *Proceedings of the ACL-2016*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling Politeness in Neural Machine Translation via Side Constraints. In *Proceedings of the NAACL-HLT-2016*. pages 35–40.

Hirotoshi Taira, Katsuhito Sudoh, and Masaaki Nagata. 2012. Zero pronoun resolution can improve the quality of je translation. In *Proceedings of the SSST-2012*. pages 111–118.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of LREC-2012*.

Yulia Tsvetkov, Chris Dyer, Lori Levin, and Archna Bhatia. 2013. Generating english determiners in phrase-based translation with synthetic translation options. In *Proceeding of the WMT-2013*. pages 271–280.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of ACL-2003*. pages 72–79.

Longyue Wang, Zhaopeng Tu, Xiaojun Zhang, Hang Li, Andy Way, and Qun Liu. 2016. A Novel Approach to Dropped Pronoun Translation. In *Proceedings of the NAACL-2016*. pages 983–993.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of EMNLP-2015*.

Joern Wuebker, Spence Green, John DeNero, Sasa Hasan, and Minh-Thang Luong. 2016. Models and Inference for Prefix-Constrained Machine Translation. In *Proceedings of the ACL-2016*. pages 66–75.

Bing Xiang, Xiaoqiang Luo, and Bowen Zhou. 2013. Enlisting the ghost: Modeling empty categories for machine translation. In *Proceedings of the ACL-2013*. pages 822–831.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the Voice of a Sentence in Japanese-to-English Neural Machine Translation. In *Proceedings of the WAT-2016*. pages 203–210.

Muyun Yang, Hongfei Jiang, Tiejun Zhao, and Sheng Li. 2006. Construct trilingual parallel corpus on demand. In *Chinese Spoken Language Processing*, Springer, pages 760–767.