Prompt-Driven Neural Machine Translation

Yafu Li[♠♡], **Yongjing Yin**[♠♡], **Jing Li**[♣], **Yue Zhang**[♡◇]
[♠] Zhejiang University
[♡] School of Engineering, Westlake University
[◇] Institute of Advanced Technology, Westlake Institute for Advanced Study
[♣] Sichuan Lan-bridge Information Technology Co., Ltd.
yafuly@gmail.com *yinyongjing@westlake.edu.cn*
judyli.5266@gmail.com *yue.zhang@wias.org.cn*

Abstract

Neural machine translation (NMT) has obtained significant performance improvement over the recent years. However, NMT models still face various challenges including fragility and lack of style flexibility. Moreover, current methods for instance-level constraints are limited in that they are either constraint-specific or model-specific. To this end, we propose prompt-driven neural machine translation to incorporate prompts for enhancing translation control and enriching flexibility. Empirical results demonstrate the effectiveness of our method in both prompt responding and translation quality. Through human evaluation, we further show the flexibility of prompt control and the efficiency in human-in-the-loop translation.

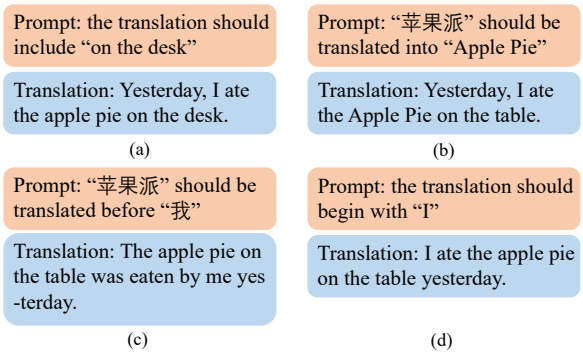

Figure 1: A Prompt-driven NMT model outputs different translations for the sentence “昨天, 我吃了桌上的苹果派。” (English: Yesterday, I ate the apple pie on the table.) based on the given prompts. One can specify phrase translations, guarantee translation positions or alter word order by feeding the system with different prompts.

1 Introduction

Neural machine translation (NMT) has achieved much performance improvement over the recent years (Vaswani et al., 2017; Edunov et al., 2018; Hassan et al., 2018; Liu et al., 2020), yet still faces various challenges such as low cross-domain robustness (Müller et al., 2020), fragility (Li et al., 2021) and lack of style flexibility (Li and Jurafsky, 2016; Shu et al., 2019). To address these issues, a line of work considers introducing constraints to the translation outputs, typically in the form of lexical constraints (Song et al., 2019; Chen et al., 2020) and style control (Sennrich et al., 2016a; Michel and Neubig, 2018; Shu et al., 2019). For example, Song et al. (2019) ensure that polysemous words are translated to their domain-specific senses in eCommerce.

Such instance-level constraint has been shown useful for improving both the translation adequacy and readability in practical applications (Song et al., 2019; Chen et al., 2020; Jwalapuram et al., 2020; Konieczny, 2021; Chen et al., 2021a). However, they are limited in being (1) model-specific and (2) constraint-specific. For instance, lexical constraints are typically integrated into a model by either modifying the decoding process (Hokamp and Liu, 2017; Post and Vilar, 2018; Chen et al., 2021a) or introducing special post-processing (Song et al., 2019; Chen et al., 2020). Style constraints are learned through data synthesization (Sennrich et al., 2016a; Niu and Carpuat, 2020) or specialized model design (Michel and Neubig, 2018). As a result, the engineering cost of accommodating and simultaneously optimizing for various constraints and styles can be high.

We consider prompt-driven neural machine translation, a general form of introducing translation constraints. The basic idea is shown in Figure 1, where a prompt-driven NMT system can accept a source input, together with an arbitrary number of instructions, and generate a target translation in accordance. Since the translation constraints are specified in texual form, we can integrate different types of control easily into the input, such as specifying the translation of a source phrase (Figure 1b), controlling word order (Figure 1c) and laying

Findings of the Association for Computational Linguistics: ACL 2022, pages 2579 - 2590

out the beginning of the target sentence (Figure 1d), in addition to the traditional lexical constraints (Figure 1a). In addition, when there are no input constraints, the NMT system should give competitive performance as a unconstrained NMT model.

Without losing generality, we consider the forms of constraints in Figure 1 in this work. Building on a standard Transformer (Vaswani et al., 2017) baseline, we consider the following research questions. *First, what is the most effective system architecture for encoding both the source sentence and the prompt?* To this end, we compare various methods including concatenating source sentences with prompts, encoding prompts using a dedicated module, and incorporating prompt representations with an attention layer. The model performance is also compared with previous work on lexical constraints, a form of constraints in Figure 1 that has been much studied in the literature. *Second, can different types of constraints be effectively trained within the same model?* To this end, we design an algorithm to automatically construct different types of prompts from a standard MT training corpus, training a model with mixed prompts. *Third, can a prompt-driven NMT system accept different number of prompts, while maintaining the same level of performance compared to a Transformer baseline without constraints?* To this question, we consider a sampling-based training strategy, where the model receives random combinations of arbitrary number of prompts or no prompt at all for each sample during training. *Fourth, can the set of flexible constraints we use serve to improve the efficiency of human-in-the-loop translation?* We deploy our prompt-drive system in a real application scenario where professional translators conduct machine translation post editing (MTPE) by using prompts.

Empirical results show that the Prompt-driven Transformer (Prompt-Transformer) responds to different prompts effectively, while giving competitive performance when used as a unconstrained NMT model. In addition, prompt-driven model outperforms previous lexical constraints methods (Song et al., 2019; Chen et al., 2021b) by a large margin. Human experiments further demonstrate the control flexibility and effectiveness of our method. Through system deployment in a practical scenario, we show that the prompt-driven NMT system achieves a trade-off between translation quality and human efficiency, as compared with full NMT or NMT with human post editing. Our code is released on https://github.com/yafuly/PromptNMT.

2 Related Work

Lexical constraint has received much attention for machine translation. Some researchers incorporate the constraints into the beam search algorithm (Hokamp and Liu, 2017; Post and Vilar, 2018), and recently Chen et al. (2021b) investigate alignment-based constrained decoding methods using attention weights. Another approach focuses on data augmentation. Song et al. (2019) and Dinu et al. (2019) create a synthetic code-switching corpus. Jon et al. (2021) augment the input sentences with lemmatized constraints to correct inflection. Chen et al. (2020) propose a lexical constraint-aware Transformer model (LeCA) by concatenating constraints and source sentence. Lexical constraints is one of the application scenarios of our method. Prompt-driven model gives strong results, while also simultaneously enables structural and style constraints with the versatility of prompts.

There has been study on controlling the global output style in MT (Mima et al., 1997; van der Wees et al., 2016; Rabinovich et al., 2017; Michel and Neubig, 2018; Sennrich et al., 2016a; Niu and Carpuat, 2020). van der Wees et al. (2016) analyze the impact of dialogue specific aspects in SMT for fictional dialogues. Rabinovich et al. (2017) employ personalized SMT models for better preservation of gender traits, and Michel and Neubig (2018) propose to adapt the bias of the output softmax to different users of an NMT system. Sennrich et al. (2016a) use target-constraint T-V annotation in NMT training to control the level of politeness. Niu and Carpuat (2020) propose a formality-sensitive NMT model taking formality levels as an extra input. Our work is similar in that the output of our model can be adaptive at inference time, but different in that the control is more fine-grained and not limited to certain styles.

Human in the loop for NMT (Turchi et al., 2017; Weng et al., 2019) has been proved effective to domain adaptation. Cheng et al. (2016) propose an interactive framework which takes two human actions: picking a critical translation error and revising the translation. Petrushkov et al. (2018) propose a simple sentence-level weighting method to integrate partial chunk-based feedback into NMT. Kreutzer et al. (2018) improve NMT with explicit and implicit user feedback collected on the ecom-

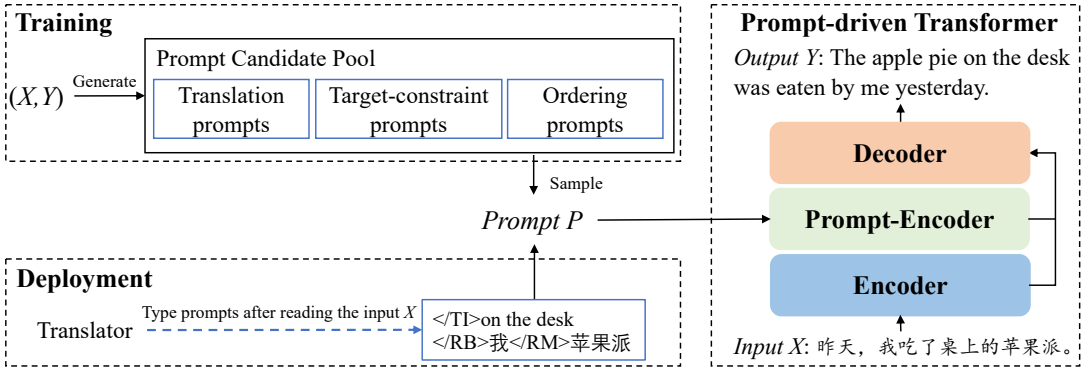

Figure 2: The overall framework of Prompt-Transformer. During training, the prompts are sampled from the prompt candidate pool, which contains all possible prompts for each sentence pair. In deployment, the translators give arbitrary prompts to control output translations according to their needs.

merce platform. Domingo et al. (2019) leverage data generated during the post-editing process. The above methods improve the performance of NMT by leveraging extra training signals from human feedback. Different from them, our method allows human to control the NMT output by training a model with mixed prompts, without the requirement of human in training.

3 Problem Definition

In neural machine translation, a set of parallel sentence pairs $D = \{(X, Y)\}$ is given where $X = (x_1, ..., x_{T_x})$ and $Y = (y_1, ..., y_{T_y})$, and the NMT systems model the conditional probability:

$$p(Y|X;\theta) = \prod_{t}^{T_y} p(y_t|y_{<t}, X;\theta), \quad (1)$$

where θ is the set of trainable parameters. We introduce prompts $P = (P_1, ..., P_N)$ to control translation, which is defined as

$$p(Y|X,P;\theta) = \prod_{t}^{T_t} p(y_t|y_{<t}, X, P;\theta). \quad (2)$$

The prompts can be general and flexible. In this paper, we consider the following three types of common prompts:

- **translation prompts** that indicate the specific translation of a source segment (Fig 1 (b)).
- **target-constraint prompts** including some specific segments that the translation must contain, begin or end with (Fig 1 (a) and (d)).
- **ordering prompts** that indicate a source segment should be translated before another source segment (Fig 1 (c)).

4 Approach

The overall architecture of our system is shown in Figure 2. In particular, we take a Transformer baseline (Section 4.1), discussing different ways to additionally encode prompt constraints (Section 4.2). We propose a sampling-based training framework (Section 4.4), with automatic methods for generating rich constraints from standard MT training instances (Section 4.3).

4.1 Transformer

The vanilla Transformer (Vaswani et al., 2017) is composed of an encoder and a decoder. The Transformer encoder has a stack of L identical multi-head self-attention layers, which takes the embedding of a source sentence X as input and outputs contextualized source representations. For the l-th encoder layer, the representations are computed as

$$H^l = \text{EncLayer}(H^{l-1}), \quad (3)$$

where H^{l-1} is the output hidden state of the $(l-1)$-th layer.

The decoder introduces a cross-attention sub-layer in each layer to attend to the source representations H^L, taking previously generated target tokens as input and generating the next token. For the l-th decoder layer, the hidden states of decoder are calculated as

$$S^l = \text{DecLayer}(S^{l-1}, H^L), \quad (4)$$

where S^{l-1} is the output of the $(l-1)$-th layer.

4.2 Prompt-driven Transformer

We investigate three different approaches to incorporate prompts into the Transformer model.

(1) Separate Encoding. A straightforward way is to introduce a Prompt Encoder that is identical to the Transformer encoder, which encodes the prompt sequence separately. We concatenate the source representations and the prompt representations as the final encoder memory for the decoder:

$$H_P^L = \text{Prompt-Encoder}(P), \tag{5}$$

$$\hat{H}_L = \text{Concat}(H^L, H_P^L), \tag{6}$$

where P is a prompt sequence.

(2) Input Augmentation. We follow Chen et al. (2020) and construct pseudo source sequences by augmenting each input source sequence with the corresponding prompt sequence:

$$\hat{X} = \text{Concat}(X, P_1, P_2, ..., P_N), \tag{7}$$

where N is the number of prompts. The augmented input $\hat{X}$ is fed into the standard Transformer.

(3) Prompt Attention. On top of the concatenation method, we can also use a dedicated prompt attention sub-layer after the cross-attention module in each decoder layer. The prompt attention takes the decoder hidden representations as queries and takes the prompt representations as keys and values to perform multi-head attention:

$$\text{PromptAttn}(S^l, H_P^L) = \text{MHA}(S^{l-1}, H_P^L, H_P^L), \tag{8}$$

where $\text{MHA}(\cdot)$ is the multi-head attention mechanism (Vaswani et al., 2017).

4.3 Training Prompt Construction

Given a parallel dataset $D = \{(X, Y)\}$, we propose an automatic method to generate prompts for each sentence pair based on word alignment, resulting in a corpus $\hat{D} = \{(X, Y, \hat{P})\}$, where $\hat{P}$ is the corresponding prompt candidate pool containing all prompts. Specifically, we train an alignment tool on a parallel corpus and obtain possibly aligned phrases. For each sentence pair, we extract all possible prompts using the aligned phrases to build the prompt candidate pool.

First, we insert pre-defined symbols between source phrase segments and the corresponding aligned target segments (e.g., "</AB> menschliche gesundheit </AM> human health") to construct **translation prompts**. Second, we append pre-defined symbols before target phrase segments to construct **target-constraint prompts**: (1) "</TB>" denotes the target sequence begins with specific segments (e.g., "</TB> we know"); (2) "</TI>" denotes the target sequence includes specific segments (e.g., "</TI> the complex science"); (3) "</TE>" denotes the target sequence ends with specific segments (e.g., "</TE> we 've experienced that ."). Third, for **ordering prompts**, we find pairs of source phrases of which the aligned target phrases appear in the opposite order in the target sequence, indicating word-reordering is involved in translating these phrases. We insert pre-defined symbols between these 2 source segments (e.g., "</RB> the apple pie </RM> on the table", meaning that "on the table" should be translated before "the apple pie" in the target language).

4.4 Training

Given $\hat{D} = \{(X, Y, \hat{P})\}$, we propose a sampling based training framework to train the prompt-driven NMT model. For each instance $(X, Y, \hat{P})$, we define whether to use prompts as a discrete Bernoulli variable $u \sim \mathcal{B}(\mu)$, where μ is a hyper-parameter (*Bernoulli ratio*) and a higher μ indicates more prompt-driven samples during training. If prompt is not used, the training objective is to maximize the log-likelihood:

$$\sum_{(X,Y)\in Batch} logp(Y|X;\theta), \tag{9}$$

where $Batch$ is a mini-batch of parallel sentence pairs.

If prompt is used, we sample a certain proportion of prompts for each prompt type from the corresponding prompt candidates without replacement. In particular, we define the proportion of the sampled prompts as a continuous random variable with a uniform distribution $\mathcal{U}(0, p_u)$, where p_u is a hyper-parameter, *uniform ratio*. A larger p_u indicates more prompts are sampled for each sentence if there are. All sampled prompts are concatenated together to form the final prompt sequence P, and the training objective is to maximize the log-likelihood defined as:

$$\sum_{(X,Y,P)\in Batch} logp(Y|X,P;\theta). \tag{10}$$

The randomness in prompts enables the model to cope with complicated situations containing different prompts and output accurate translations without prompts as well.

Model	# params	BLEU		ResR
		w/o prompts	w/ prompts	
Transformer-IWSLT	36.74M	34.78	34.78	-
Prompt Encoder	43.05M	34.27	53.73	92.08
Param-share Prompt Encoder	36.74M	34.44	54.83	93.30
Prompt Enc & Prompt Attention	49.36M	34.28	53.79	92.20
Param-Share Prompt Enc & Prompt Attn	43.06M	34.04	55.06	94.35
Input Augmentation	36.74M	33.69	56.10	95.19

Table 1: Performance of different prompt-feeding methods on IWSLT'14 De-En.

5 Experimental Settings

Setup. As a preliminary experiment, we use a small size dataset IWSLT'14 De→En to investigate the effectiveness of our model under different settings. We use the Moses tokenizer[1] and apply BPE (Sennrich et al., 2016b) with 10,000 merge operations on the merged corpus of both side. For large-scale test, we extend our method to WMT'17 En→Zh, which contains 20.6M sentence pairs after preprocessing. We use Moses tokenizer to tokenize English side and jieba segmenter[2] to tokenize Chinese side. We apply BPE with 55,000 operations on the concatenated corpus and obtain a shared vocabulary for both sides. We use fast_align (Dyer et al., 2013) to obtain word alignment, based on which we apply the algorithm in Section 4.3 to generate prompts and build the prompt candidate pool. Data statistics is presented in Appendix A. We implement the Transformer baseline and Prompt-Transformer based on THUMT (Tan et al., 2020). We use iwslt_de_en for IWSTLT'14 De→En and transformer_base for WMT'16 En→Zh. The default Prompt Encoder consists of 3 Transformer layers. We use Adam (Kingma and Ba, 2015) to optimize the network with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The default Bernoulli and uniform ratios are set as 0.3 and 0.35, respectively. For inference, we set the beam width as 5 and length penalty as 0.6. Details are presented Appendix B.

Evaluation Metrics. We use both automatic and human evaluation to measure the performance of our prompt-driven NMT model, taking commonly-used BLEU scores (Papineni et al., 2002) to measure translation quality automatically. For fair comparison with previous work, we use multi-bleu.perl for De-En and sacreBLEU (Post, 2018) for En-Zh[3]. In addition, we use Response Rate (ResR) to quantify how the model responses to the given prompts, which is defined as the percentage of prompts being correctly responded. Specifically, for translation prompts, ResR denotes the ratio of prompt translations that appear in the sentence translation; for target-constraint prompts, ResR measures the ratio of prompts that exist at the beginning of, at the end of or in the translation accordingly; for ordering prompts, ResR is calculated as the ratio of translations that satisfy the word ordering information induced by the prompts.

For human evaluation, we follow Knight (2000) and ask professional translators to assign *adequacy* and *fluency* scores for each translation ranging from one to five. The five point scale for adequacy indicates how much of the meaning expressed in the reference translation is also expressed in a hypothesis translation: 5 = All, 4 = Most, 3 = Much, 2 = Little, and 1=None. The five point scale for fluency indicates how fluent the translation is: 5 = Flawless, 4 = Good, 3 = Non-native, 2 = Disfluent, and 1 = Incomprehensible.

We investigate the effectiveness of our method in the context of automatic evaluation in Section 6, where prompts are constructed towards reference translation. In Section 7, we conduct human evaluation to demonstrate the control flexibility of the Prompt-driven NMT system. Finally, in Section 8 we show an application of the method in the context of human-in-the-loop translation.

6 Experiments on the Model Design

We evaluate models under two test scenarios using IWSLT'14 De-En: inference without prompt and inference with prompt. The former is the same as the vanilla machine translation setting and is evaluated using BLEU score. For the latter, we also evaluate the model's effectiveness on responding to prompts by calculating ResR. We apply sampling strategy same to training and run on the test set once to build a deterministic prompt sets.

[1]https://github.com/moses-smt/
[2]https://github.com/fxsjy/jieba
[3]Sig: BLEU+c.mixed+l.en-zh+#.1+s.exp+tok.zh+v.1.5.1

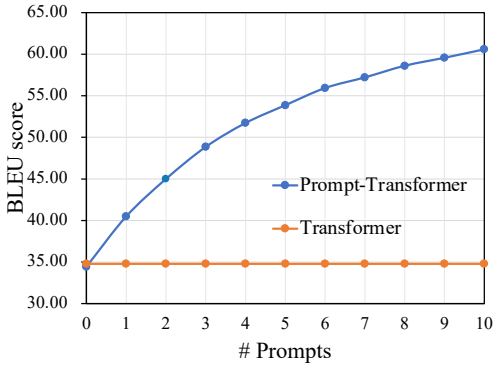

Figure 3: BLEU scores with respect to the number of prompts during inference.

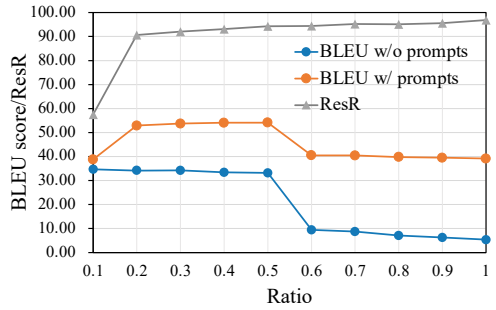

Figure 4: ResR and BLEU scores with respect to the Bernoulli ratio during training.

Number of prompts during decoding. We investigate how prompts improve translation performance by feeding different number of prompts during decoding. Specifically, we randomly select certain number of prompts from the prompt candidate pool and construct test prompts accordingly. The results are shown in Figure 3. Prompt-Transformer further achieves higher BLEU scores when there are more prompts. Given as many as 10 prompts, the BLEU reaches 60.59. We also investigate how the sampling ratio affects decoding performance, which is discussed in Appendix C.

Robustness to different prompts. We explore how the model behaves under different prompt sets, by fixing the sampling ratios but varying the seed for prompt sampling. We conduct experiments with 10 seeds, under which the model receives different prompts for translation, calculating the mean and standard deviation of BLEU scores and ResR over each seed. For each sentence, the model is provided with 1 to 8 sampled prompts. The model achieves a average BLEU score of 54.79 with a standard deviation of 0.17, and an average of 92.82 with a standard deviation of 0.14 for ResR, demonstrating that the model is stable for flexible types of prompt combinations.

Influences of model architecture. Based on the modules in Section 4.2, we compare different model architectures to incorporate prompts using a fixed prompt seed. As shown in Table 1, all prompt-driven models obtain higher BLEU scores over Transformer when provided with prompts. *Input augmentation* achieves the highest ResR, but suffers from larger performance deterioration without prompts. For the prompt encoding method, we find that reusing the sentence encoder as the prompt encoder (*Param-share Prompt Encoder*) achieves higher ResR than introducing extra parameters (*Prompt Encoder*). We attribute this pattern to the better generalization ability of the reused encoder in *Param-share Prompt Encoder*. The effects of prompt encoder depth is discussed in Appendix D. For incorporating prompt representations, introducing Prompt Attention (*Prompt Enc & Prompt Attention* and *Param-Share Prompt Enc & Prompt Attn*) is beneficial for responding effectiveness, compared with concatenating source and prompt representations for cross-attention. Overall, *Param-share Prompt Encoder* gives a balance between BLEU in unprompted cases and the response rate, without introducing extra parameters. We thus choose the model for the other experiments.

Number of prompts during training. The sampling strategy in Section 4.4 can affect the performance. We investigate how varying the Bernoulli ratio during training affects the model performance. The Bernoulli ratio indicates how many of samples in the train set are driven by prompts. For example, a Bernoulli ratio of 0.3 denotes 65.7% of the training samples are provided with prompts. The result is shown in Figure 4. We can observe that ResR grows steadily with the increasing ratio during training. The model gives a low ResR with a Bernoulli ratio of 0.1, as there are limited samples for the model to capture prompt patterns. Despite the increasing ResR, there is a sharp decline on BLEU scores when the ratio exceeds 0.5. This is because high Bernoulli ratios indicate almost all training samples are prompted (e.g., a ratio of 0.7 denotes 97.3% of training samples are provided with prompts). Therefore, the model learns to output translations by over reliance on prompts, but fails to build correspondence between source and target languages. Thus it is important to balance the learning of translation and receiving prompts.

Prompts	Translations
Null	在庭审中, 双方就王志安是否侵犯了兰玉峰的名誉权进行了辩论。 (English: in the court hearing , the two sides launched a debate on whether wang zhian violated the reputation right of lan yuefeng.)
</AB> lan yuefeng </AM> Lan Yuefeng	在庭审中, 双方就王志安是否侵犯了Lan Yuefeng 的名誉权进行了辩论。 (English: in the court hearing , the two sides launched a debate on whether wang zhian violated the reputation right of Lan Yuefeng.)
</TB> 双方	双方 在庭审中争论王志安是否侵犯了兰玉峰的名誉权。 (English: the two sides argued in the court hearing whether wang zhian violated the reputation right of lan yuefeng.)
</RB> wang zhian </RM> argued	双方在庭审中争辩 说,王志安 是否侵犯了兰玉峰的名誉权。 (English: the two sides argued in the court hearing, whether wang zhian violated the reputation right of lan yuefeng.)
</TB> 在庭审中 </RB> wang zhian </RM> argued	在庭审中 , 双方争辩 说王志安是否侵犯了兰玉峰的名誉权。 (English: in the court hearing, the two sides argued whether wang zhian violated the reputation right of lan yuefeng.)
</AB> lan yuefeng </AM> Lan Yufeng </TB> 在庭审中 </RB> wang zhian </RM> argued	在庭审中 , 双方争辩 说王志安 是否侵犯了Lan Yufeng 的名誉权。 (English: in the court hearing, the two sides argued whether wang zhian violated the reputation right of Lan Yufeng.)

Table 2: Given different prompts, Prompt-Transformer generates different translations for the sentence "in the court hearing , the two sides argued whether wang zhian violated the reputation right of lan yuefeng.".

Model	BLEU		CSR	ResR
	w/o P	w/ P		
TF-IWSLT	34.78	-	-	-
Code-Switch	33.88	37.15	93.69	90.21
LeCA	34.66	37.10	89.32	82.97
Prompt-TF	34.44	38.30	95.75	94.26

Table 3: Prompt-driven Transformer for lexical constraints on IWSLT'14 De-En. P denotes 'prompts'.

Model	BLEU		ResR
	w/o prompts	w/ prompts	
TF-Base	34.06	34.06	-
Prompt-TF	33.88	48.93	91.80

Table 4: Performance on WMT'17 En-Zh.

Comparison with existing work on lexical constraints. Among the types of prompts we accommodate, lexical constraints have been investigated by existing work. We compare our method with two typical methods, i.e., CodeSwitch (Song et al., 2019) and LeCA (Chen et al., 2021b). Following Song et al. (2019) and Chen et al. (2021b), the copy success rate (CSR) is also calculated, which is the percentage of successfully generated tokens in constraints, differing from ResR which is the ratio of correctly responded prompts (i.e., phrases for lexical constraints). Compared with CodeSwitch, Prompt-Transformer maintains better performance without prompts, while also achieves a higher score of CSR and ResR. Although LeCA is slightly better in terms of BLEU without prompts, Prompt-Transformer outperforms LeCA by a large margin in terms of CSR and ResR. Performance in lexical constraints further demonstrates the effectiveness of our method for controlling translation and meanwhile maintaining performance without prompts.

Experiments on WMT. For a large scale test, we apply Prompt-Transformer on the WMT'17 En→Zh dataset. Based on the preliminary experiments, we choose the *Param-share Prompt Encoder* architecture. As shown in Table 4, Prompt-Transformer gives an improved BLEU with prompts (48.93 vs. 34.06) and a ResR of 91.80, verifying the scalability of the proposed method on large-scale datasets. We use this model for experiments in Section 7 and Section 8.

7 Experiments on Prompts

We evaluate how model responds to prompts in practical scenarios, where no "gold-standard" references are given. We sample 100 source sentences from the WMT'17 En-Zh test set and ask 2 professional translators to assign each sentence with two different prompt groups, each of which includes at least one type of prompts. In particular, for constructing translation prompts, the translators are asked to give a source segment two different valid translations (e.g., "translation-segment-1" or "translation-segment-2"); for constructing target-constraint prompts, the translators should choose two different ways to prompt the model; for constructing ordering prompts, the translators provide two opposite orderings (e.g., "source-segment1" should be translated before and after "source-segment2", respectively). The model is expected to output two *different* and *correct* trans-

lations corresponding to the two prompt groups, respectively. We ask 3 professional translators to evaluate the *ResR* and *translation quality* based on the adequacy and fluency metrics in Section 5.

The system achieves ResR scores of 89.80, 94.74, 90.20 for translation prompts, target-constraint prompts, and ordering prompts, respectively, showing the effectiveness of our proposed model on responding to human prompts. The system obtains a competitive performance compared to the unprompted baseline in tuns of both adequacy (3.49 vs. 3.40) and fluency (3.24 vs. 3.31), demonstrating that our system can enable flexible translation style and maintain translation quality at the same time.

Table 2 shows a case study, where the system responds to different types of prompts and their combinations accurately given the same source sentence. Moreover, the system generates translations with different styles under the target-constraint prompts and ordering prompts. For instance, with the prompt "</TB> 双方" (English: </TB> the two sides), the system translates the word "argued" to "争论" (English: argued) instead of "进行了辩论" (English: launched a debate) in the unprompted case. A similar pattern can be observed when the system receives the ordering prompt "</RB> wang zhian </RM> argued", which indicates that the word "argued" should be translated before "wang zhian" in Chinese.

8 Human-in-the-loop Translation

Machine translation post-editing (MTPE) is widely used by translation companies to improve efficiency as well as ensure translation quality. Studies show that conducting post-editing over high-quality MT can increase the productivity of professional translators compared to manual translation 'from scratch' (Guerberof, 2009; Plitt and Masselot, 2010). However, MTPE still can be expensive in heavy involvement of human efforts in editing. To alleviate human labor, Prompt-driven methods can be used for a better trade-off between translation quality and efficiency.

To verify our hypothesis, we ask professional translators to compare two methods for editing on MT translations: the traditional MTPE or giving prompts based on MT translations (MTPrompt). We compare MT, MTPE and MTPrompt based on time efficiency and translation quality. MT refers to use machine translations without editing. For

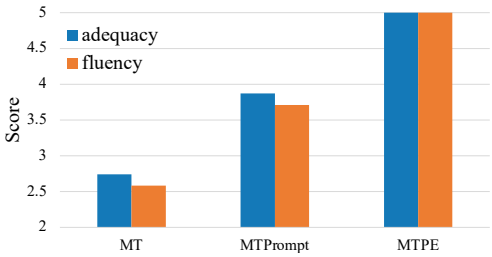

Figure 5: Translation quality based on adequacy and fluency.

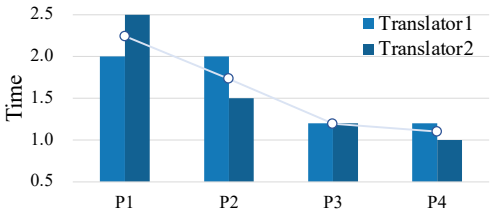

Figure 6: Time cost (hours) for MTPrompt with respect to the round of MTPrompt.

MTPE, translators are required to edit translations output by the WMT-trained Transformer baseline in Section 6. For MTPrompt, translators are required to observe output translation errors and give prompts to correct them. More details are presented in Appendix E.

The translation quality is presented in Figure 5. MTPE achieves full marks on both adequacy and fluency, whereas the scores for MT translations are on average around 2.5. Translations with prompt obtain substantial improvement over MT translations, with both the adequacy and fluency scores being close to 4 (i.e., the translations cover most meaning and also have good fluency).

In terms of speed, the average time spent on MTPE is 3.75 hours, which is stable for more batches since the translators have strong experience in MTPE. In contrast, the time cost can be lower as they conduct more MTPrompt actions. We ask two translators to conduct multiple rounds of MTPrompt edit, with each round containing 50 translations. The time cost for each round is shown in Figure 6. We can observe that as the translators get familiar with the MTPrompt mode, they become more efficient in giving prompts. The fastest batch costs an average of 1.1 hours for MTPrompt, which is 2.4 times more efficient than MTPE, and meanwhile translation quality is maintained (adequacy: 3.87 vs. 3.84 and fluency: 3.71 vs. 3.63).

9 Conclusion

We proposed a prompt-driven Transformer model to incorporate flexible constraints on translation. Under a sampling-based training framework, the model learned prompt responding effectively and achieved competitive performance compared with both the unconstrained baseline and existing work on lexical constraints. Human experiments further demonstrated that Prompt-Transformer was able to respond to various combinations of prompts accurately, and generate versatile translations. Through deployment in an application scenario, we showed that our system could serve to improve the efficiency of human-in-the-loop translation.

10 Ethics Consideration

As mentioned, we collected our data from IWSLT and WMT that all are public to academic use, and they contain no sensitive information. The legal advisor of our institute confirms that the sources of our data are freely accessible online without copyright constraint to academic use. Our human experiments (Section 7 and Section 8) involves manual annotation. Annotators were asked to give prompts, post-edit machine translation and evalaute transations, which do not involve any personal sensitive information. We hired 4 annotators who have degrees in English Linguistics or Applied Linguistics. Before formal annotation, annotators were asked to annotate a few samples randomly extracted from the dataset, and based on average annotation time we set a fair salary (i.e., 30 dollars per hour) for them. During their training annotation process, they were paid as well.

Acknowledgements

Yue Zhang is the corresponding author. We thank all reviewers for their insightful comments. This publication has emanated from research conducted with the financial support of the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant Number 2022SDXHDX0003. This work is also under a grant from Lan-bridge Information Technology Co., Ltd. We thank colleagues from Lan-bridge for examining data and evaluating results.

References

Guanhua Chen, Yun Chen, and Victor O. K. Li. 2021a. Lexically constrained neural machine translation with explicit alignment guidance. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12630–12638.

Guanhua Chen, Yun Chen, and Victor O. K. Li. 2021b. Lexically constrained neural machine translation with explicit alignment guidance. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12630–12638.

Guanhua Chen, Yun Chen, Yong Wang, and Victor O. K. Li. 2020. Lexical-constraint-aware neural machine translation via data augmentation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3587–3593.

Shanbo Cheng, Shujian Huang, Huadong Chen, Xin-Yu Dai, and Jiajun Chen. 2016. PRIMT: A pick-revise framework for interactive machine translation. In *Proc. of NAACL-HLT*, pages 1240–1249.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proc. of ACL*, pages 3063–3068.

Miguel Domingo, Mercedes García-Martínez, Álvaro Peris, Alexandre Helle, Amando Estela, Laurent Bié, Francisco Casacuberta, and Manuel Herranz. 2019. Incremental adaptation of NMT for professional post-editors: A user study. In *Proceedings of Machine Translation Summit XVII: Translator, Project and User Tracks*, pages 219–227.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proc. of NAACL-HLT*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proc. of EMNLP*, pages 489–500.

Ana Guerberof. 2009. Productivity and quality in MT post-editing. In *Beyond Translation Memories: New Tools for Translators Workshop*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proc. of ACL*, pages 1535–1546.

Josef Jon, João Paulo Aires, Dusan Varis, and Ondřej Bojar. 2021. End-to-end lexically constrained machine translation for morphologically rich languages. In *Proc. of ACL*, pages 4019–4033.

Prathyusha Jwalapuram, Shafiq Joty, and Youlin Shen. 2020. Pronoun-targeted fine-tuning for NMT with hybrid losses. In *Proc. of EMNLP*, pages 2267–2279.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Kevin Knight. 2000. Statistical machine translation. In *Proceedings of the Fourth Conference of the Association for Machine Translation in the Americas: Tutorial Descriptions*.

Jakub Konieczny. 2021. Training of neural machine translation model to apply terminology constraints for language with robust inflection. In *Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems, Online, September 2-5, 2021*, pages 233–234.

Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can neural machine translation be improved with user feedback? In *Proc. of NAACL-HLT*, pages 92–105.

Jiwei Li and Dan Jurafsky. 2016. Mutual information and diverse decoding improve neural machine translation. *CoRR*, abs/1601.00372.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. On compositional generalization of neural machine translation. In *Proc. of ACL*, pages 4767–4780.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. 2020. Very deep transformers for neural machine translation. *CoRR*, abs/2008.07772.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proc. of ACL*, pages 312–318.

Hideki Mima, Osamu Furuse, and Hitoshi Iida. 1997. Improving performance of transfer-driven machine translation with extra-linguistic informatioon from context, situation and environment. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29, 1997, 2 Volumes*, pages 983–989.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164.

Xing Niu and Marine Carpuat. 2020. Controlling neural machine translation formality with synthetic supervision. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8568–8575.

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

Pavel Petrushkov, Shahram Khadivi, and Evgeny Matusov. 2018. Learning from chunk-based feedback in neural machine translation. In *Proc. of ACL*, pages 326–331.

Mirko Plitt and François Masselot. 2010. A productivity test of statistical machine translation post-editing in a typical localisation context. *Prague Bull. Math. Linguistics*, pages 7–16.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proc. of NAACL-HLT*, pages 1314–1324.

Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. 2017. Personalized machine translation: Preserving original author traits. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1074–1084.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proc. of NAACL-HLT*, pages 35–40.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proc. of ACL*, pages 1715–1725.

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *Proc. of ACL*, pages 1823–1827.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proc. of NAACL-HLT*, pages 449–459.

Zhixing Tan, Jiacheng Zhang, Xuancheng Huang, Gang Chen, Shuo Wang, Maosong Sun, Huanbo Luan, and Yang Liu. 2020. THUMT: An open-source toolkit for neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine*

Translation in the Americas (Volume 1: Research Track), pages 116–122.

Marco Turchi, Matteo Negri, M. Amin Farajian, and Marcello Federico. 2017. Continuous learning from human post-edits for neural machine translation. *Prague Bull. Math. Linguistics*, pages 233–244.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2016. Measuring the effect of conversational aspects on machine translation quality. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2571–2581.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Rongxiang Weng, Hao Zhou, Shujian Huang, Lei Li, Yifan Xia, and Jiajun Chen. 2019. Correct-and-memorize: Learning to translate from interactive revisions. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5255–5263.

A Data Statistics

Dataset	# sents	avg. Tr	avg. Tc	avg. O
IWSLT	160,239	41.28	41.56	0.38
WMT	20,616,247	34.69	34.69	18.24

Table 5: Data statistics with the right 4 columns accordingly denoting number of sentences, average number of translation prompts, target-constraint prompts and ordering prompts for each sentence.

B Experiment Details

We implement the Transformer baseline and Prompt-Transformer based on THUMT (Tan et al., 2020). Except for the prompt encoding modules, Prompt-Transformer shares the same settings with the Transformer baseline. The prompt encoder layer shares the same setting with the vanilla Transformer encoder layer, and the prompt attention module is the same as the Transformer cross-attention module. For IWSLT'14 De→En, we use the iwslt_de_en setting with dropout ratio 0.3. For WMT'16 En→Zh, we use the transformer_base setting with a dropout of 0.1. We use the Adam (Kingma and Ba, 2015) to optimize the network with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The batch szie for training De→En models is 4,096 and 32,768 for En→Zh models. The default Bernoulli and uniform ratio is set as 0.3 and 0.35 respectively. For inference, we set the beam width as 5 and length penalty as 0.6.

C Effects of Uniform Ratio during Decoding

We investigate how prompts improve translation performance, by using the same sampling strategy during training but setting the Bernoulli ratio to 1, so that the number of prompts is only determined by the uniform ratio (Section 4.4). By varying the uniform ratio, the model receives different number of prompts for each sentence. The results are shown in Figure 7. We can observe that Prompt-Transformer behaves similarly to the Transformer baseline when the uniform ratio is 0, i.e., all sentences are translated without prompts. The translation performance is improved in a large degree when the uniform ratio is as small as 0.05. Prompt-Transformer further achieves higher BLEU scores when there are more prompts. With all prompts (0.35 ratio), the BLEU reaches 54.88, 20.06 higher than the baseline of 34.78.

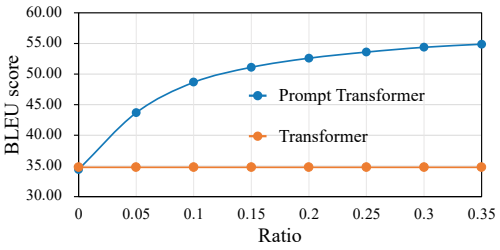

Figure 7: BLEU scores with respect to the uniform ratio during inference.

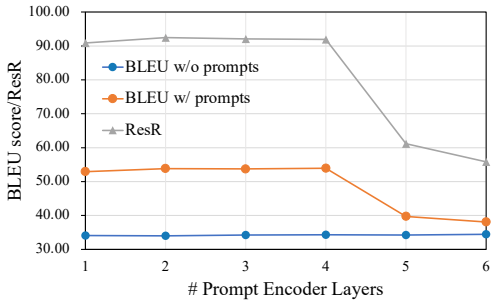

Figure 8: ResR and BLEU scores with respect to the number of prompt encoder layers.

D Effects of Prompt Encoder Depth

We investigate how the depth of the prompt encoder affects model performance. The results are shown in Figure 8. We can observe that the model performs steadily well with a prompt encoder of one to four Transformer layers. However, the ResR and BLEU score with prompts decrease sharply when the depth grows to 5 layers. This can be because that too deep prompt encoders overfit to the small scale MT dataset and thus fail to generalize to unseen prompts robustly.

E Prompt in Human-in-the-loop Translation

We sample 100 sentences from the WMT'17 En-Zh test set and ask 2 professional translators to conduct MTPE and MTPrompt on the corresponding translations. The first translator is asked to perform MTPE on the first 50 sentences and MTPrompt on the other 50 sentences, whereas the second translator is asked to do the other way around. They are required to record the time they spend with both methods. Then we ask 3 translators to evaluate translations based on adequacy and fluency mentioned in Section 5 and calculate average scores respectively.