Adaptive Prompt Routing for Arbitrary Text Style Transfer with Pre-trained Language Models

Qingyi Liu[1], Jinghui Qin[2], Wenxuan Ye[3], Hao Mou[4], Yuxuan He[4], Keze Wang[1]*

[1]Sun Yat-sen University
[2]Guangdong University of Technology
[3]X-Era AI Co., Ltd.
[4]Datastory
{scape1989,nbvincentelite,kezewang}@gmail.com, {mouhao, heyuxuan}@datastory.com.cn, liuqy95@mail2.sysu.edu.cn

Abstract

Recently, arbitrary text style transfer (TST) has made significant progress with the paradigm of prompt learning. In this paradigm, researchers often design or search for a fixed prompt for any input. However, existing evidence shows that large language models (LLMs) are prompt-sensitive and it is sub-optimal to apply the same prompt to any input for downstream TST tasks. Besides, the prompts obtained by searching are often unreadable and unexplainable to humans. To address these issues, we propose an **A**daptive **P**rompt **R**outing (APR) framework to adaptively route prompts from a human-readable prompt set for various input texts and given styles. Specifically, we first construct a candidate prompt set of diverse and human-readable prompts for the target style. This set consists of several seed prompts and their variants paraphrased by an LLM. Subsequently, we train a prompt routing model to select the optimal prompts efficiently according to inputs. The adaptively selected prompt can guide the LLMs to perform a precise style transfer for each input sentence while maintaining readability for humans. Extensive experiments on 4 public TST benchmarks over 3 popular LLMs (with parameter sizes ranging from 1.5B to 175B) demonstrate that our APR achieves superior style transfer performances, compared to the state-of-the-art prompt-based and fine-tuning methods. The source code is available at https://github.com/DwyaneLQY/APR

Introduction

Text style transfer (TST) is a task in natural language generation that falls under controllable text generation (Hu et al. 2017). Its objective is to alter the style attributes of the text while preserving its style-independent content and ensuring smooth expression. The textual style includes different attributes within a text, such as emotions, politeness, humor, and formality. TST is widely used in application scenarios, e.g., data augmentation (Kaushik, Hovy, and Lipton 2020), personalized dialogues (Zheng et al. 2021), and text detoxification (Nogueira dos Santos, Melnyk, and Padhi 2018).

The main challenge of TST is the scarcity of parallel data. Most of the previous methods train models from scratch in an unsupervised manner, limiting their transfer performances. Recent research shows that fine-tuning pre-trained language models on style transfer datasets can achieve better transfer performance (Krishna, Wieting, and Iyyer 2020; Liu, Neubig, and Wieting 2021). However, these methods still suffer from limited task data and expensive training /

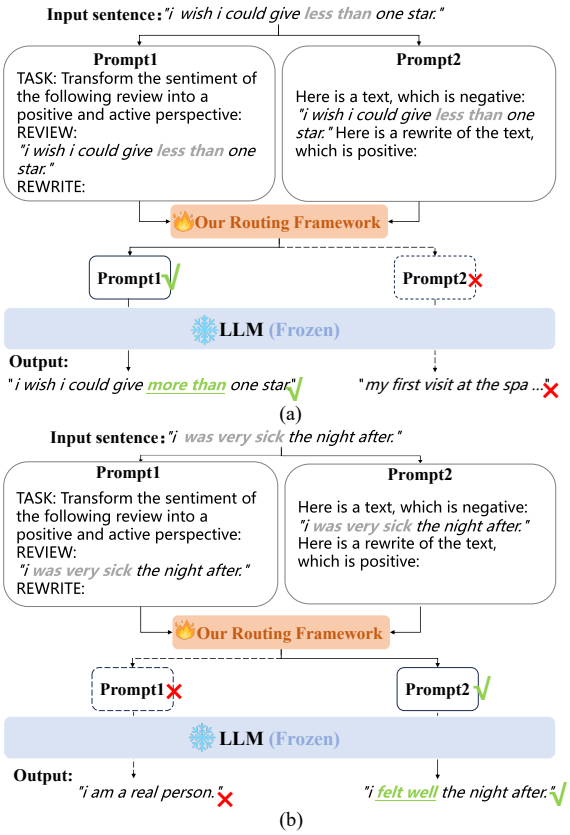

Figure 1: Illustration of results from negative-to-positive sentiment transformation using GPT-2-XL with distinct task-specific prompts for two input sentences sampled from Yelp. For different input sentences, our routing framework can adaptively select the prompt yielding a better style transformation.

*Corresponding author.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

maintaining costs of separate models for different styles.

To tackle this problem, researchers explore prompt-based methods for TST, as well-designed prompts can elicit the zero-shot / few-shot generalization capability of large language models (LLMs) in downstream tasks (Brown et al. 2020; Wei et al. 2022; Wang et al. 2022; Liu et al. 2023). Promising results have been obtained in arbitrary TST by using manual design prompt templates, as demonstrated by Reif et al. (2022) and Suzgun, Melas-Kyriazi, and Jurafsky (2022). However, designing a suitable prompt manually is still challenging. Deng et al. (2022) proposed a discrete prompt search method based on reinforcement learning, and their experiments show that a fixed prompt obtained by automatic prompt searching can usually achieve better style-transfer performance.

However, a fixed prompt from manual design or automatic searching is suboptimal for TST since the data are linguistically diverse and LLMs are prompt-sensitive (Zhao et al. 2021; Lu et al. 2021; Mishra et al. 2022; Zhu et al. 2023). As shown in Figure 1, different sentences require different prompts to achieve optimal style transformation. For the first sentence in Figure 1(a), the LLM with Prompt 1 generates a better result while the LLM with Prompt 2 generates better for the second sentence in Figure 1(b). Besides, although automatic prompt searching methods that generate a prompt from training data globally can perform better than manually designed prompts, the searched prompts are not human-readable, making them hard to explain and adaptable under practical scenarios. Therefore, it is necessary to select different suitable and human-readable prompts for different input texts for more accurate style transfer.

To address the above-mentioned issues, we propose an **A**daptive **P**rompt **R**outing (APR) framework to automatically select suitable, human-readable prompts for various input sentences to further stimulate the style-transfer performance of the LLMs. First, we construct a set of diverse and human-readable candidate prompts for the target style-transfer task. It includes several manually designed seed prompts with different degrees of detail about the task and their paraphrased variants generated via the LLM (e.g., GPT-3.5). These prompts are crucial in helping us identify suitable prompts for a wide range of input sentences. Next, we train a prompt routing model on a small amount of artificial training data, which is based on the set of prompts and raw text from original style transfer datasets. This training process is done in a supervised manner, with prompts labeled using an evaluation-related style-transfer score function. This allows the prompt router to select the optimal prompt for each input sentence during testing. Finally, we choose the highest-quality output from multiple options generated by the LLM, all guided by the adaptively routed optimal prompt. This ensures that our APR achieves the best possible style-transfer quality.

The **main** contributions of this work are two-fold: i) To the best of our knowledge, our APR framework is first to adaptively route prompts from a diverse and human-readable prompt set for various input texts and given styles in arbitrary TST. Besides, our APR is lightweight and training-efficient, making it easily adaptable to LLMs at different scales; ii) We analyze the limitations of existing prompt-based approaches for arbitrary TST and highlight the importance of selecting suitable prompts for different input sentences while ensuring that the prompts are easy for humans to understand. Extensive experiments on 4 public TST datasets over 3 popular LLMs (with parameter sizes ranging from 1.5B to 175B) demonstrate the effectiveness and superiority of our proposed APR.

Related Work

Prompt Learning Prompt learning has become a popular paradigm for stimulating the powerful generalization ability of large language models (LLMs) to solve various downstream tasks (Brown et al. 2020; Sanh et al. 2021; Wei et al. 2022; Wang et al. 2022). However, LLMs are highly prompt-sensitive (Zhao et al. 2021; Lu et al. 2021; Mishra et al. 2022; Zhu et al. 2023), making it challenging but necessary to find suitable prompts (Liu et al. 2023). While prompt engineering is a common choice (Petroni et al. 2019; Brown et al. 2020; Jiang et al. 2020; Schick and Schütze 2021; Gao, Fisch, and Chen 2020), it depends heavily on expert knowledge and may not guarantee optimal performance. Soft prompts (Lester, Al-Rfou, and Constant 2021; Gu et al. 2021; Li and Liang 2021; Liu et al. 2022, 2021), by optimizing continuous vectors through backpropagation tend to perform better but are expensive to optimize and challenging to explain. Another option is to optimize a fixed discrete prompt directly (Shin et al. 2020; Deng et al. 2022), which searches for tokens in the vocabulary space, but these prompts still lack semantic meaning and are difficult to understand. Recently, researchers used low perplexity to select human-readable prompts (Gonen et al. 2022). However, this approach still provides a fixed prompt for any input.

Text Style Transfer The TST methods are often unsupervised due to the absence of parallel data for supervised training (Xu et al. 2012; Zhang, Zhao, and LeCun 2015; Rao and Tetreault 2018). However, previous unsupervised approaches, such as training TST models from scratch (Shen et al. 2017; Li et al. 2018; Luo et al. 2019; John et al. 2019) or fine-tuning pre-trained models (Krishna, Wieting, and Iyyer 2020; Liu, Neubig, and Wieting 2021), have limited performance and require large amounts of training data. Recent studies have shown that prompt-based methods, leveraging the powerful generalization ability of LLMs, can make significant progress on arbitrary TST in a few-shot/zero-shot manner. Reif et al. (2022) improved LLM's zero-shot performance for arbitrary TST by using task-related exemplars; Suzgun, Melas-Kyriazi, and Jurafsky (2022) proposed a reranking method to select a high-quality output from the LLM to improve performance. They both used a fixed manually-designed prompt template to guide the LLM. Meanwhile, (Luo et al. 2023) proposed a prompt-based editing method that is more controllable, yet more time-consuming than autoregressive generation.

Preliminary

In the arbitrary TST task, an input sentence x with an arbitrary source style s can be denoted as x^s and a required tar-

get style is denoted as t. This task aims to transfer sentence x^s from source style s to target style t while maintaining its style-independent content. Besides, the output of style transfer is expressed as y^t. Since we mainly explore the prompt-based method for TST, we take an off-the-shelf LLM $G(\cdot)$ (e.g., GPT-2-XL) as our TST generator following the setting of prompt learning. To obtain a target text y^t, we first fill the source text x^s into a textual string prompt q, which provides sufficient style transformation information to the LLM for performing TST. The combined text of the source text x^s and the prompt q is denoted $q||x^{s_1}$. Subsequently, we feed it into LLM $G(\cdot)$ to generate the target text y^t by sampling from the output distribution $G(p||x^s)$.

Adaptive Prompt Routing

We introduce our **A**daptive **P**rompt **R**outing (APR) framework, which adaptively routes suitable prompts for the input sentences in the source style from a set of diverse and human-readable prompts. The overview of the inference phase in APR is shown in Figure 2. APR relies on two key components: a set of diverse and human-readable prompts and a prompt router that selects a suitable prompt for each input sentence. We manually design a few seed prompts and paraphrase them via GPT3.5 to get the entire set of candidate prompts. Then, we annotate each candidate prompt according to sampled raw texts from the origin dataset. Based on the annotated prompt data, we train our prompt router in a supervised manner. During testing, for a given input sentence, we choose the prompt with the highest probability of being optimal according to our prompt router and feed it with the input sentence to the LLM to perform target style transfer. Finally, we obtain the generated output by reranking the multiple sampled outputs of the LLM.

Diverse and Human-Readable Prompts

To increase the probability of finding more suitable prompts for different input texts, we construct a diverse and human-readable prompt set for the target transformation. Our procedure, similar to (Gonen et al. 2022), involves two steps: manually designing a few seed prompts describing the target transformation and obtaining a more diverse prompt set with these seed prompts and paraphrased variants via GPT3.5.

Specifically, for a given style transfer task, e.g., transforming sentences from *informal* to *formal*, we need to write some natural language instructions about this informal-to-formal transfer task as our seed prompts. (Reif et al. 2022) and (Suzgun, Melas-Kyriazi, and Jurafsky 2022) provided an example of manual style-transfer prompt. They take a sentence rewrite type instruction as their style-transfer prompt template, like *"Here is a text: which is informal* $[x]$ *Here is a rewrite of the text, which is formal: "*, where $[x]$ is the source text. We include it as one of our seed prompts, along with prompts inspired by the effective prompt framing technique (Mishra et al. 2022). These seed prompts can provide different degrees of style-transfer details. For example, we write a prompt like *"Please complete the following sentence, correcting any spelling error, grammar error, or informal expressions in the* $[x]$*"*. This type of prompt might benefit the LLM to learn how to make sentences more formal, such as "*correcting spelling errors*", instead of simply saying "*... Here is a rewrite of the text, which is formal:*".

After collecting a few seed prompts, we use GPT-3.5 to generate a larger and more diverse set of prompts using the paraphrasing instruction. The resulting prompts have similar meanings but different expressions and formats. Our paraphrase instruction is "*Suppose you are a prompting engineer. Now, I need some prompts to ask GPT 3.5 to* $[req]$. *Here is an example prompt* $[seed]$ *Please paraphrase it from 3 to 5 more prompts that are of the same meaning but more diverse in types, such as QA, dialogue, and so on.*", where $[req]$ is the requirement or description of the target style-transfer task, and $[seed]$ is one of the seed prompts. Benefiting from the powerful language ability of GPT-3.5, the generated prompts are often human-readable. To ensure readability, we manually filter these prompts again. After completing the previous two steps, we can obtain a candidate prompt set Q consisting of seed prompts and their paraphrased variants for specific style transfer tasks. For each experimental style-transfer task, $|Q|$ is set to 20 unless specified otherwise (see our supplementary file for details). These candidate prompts are human-readable and insufficiently diverse for the task description, prompt type, and expression.

Prompt Router

After constructing the diverse and human-readable prompt set, we develop an LLM-specific prompt router to route suitable prompts for different input texts adaptively. Here, we first describe a basic yet important score function for measuring the generation quality, which can be applied to guide the training of the prompt router and the output selection at the LLM inference stage. Then, we introduce the prompt router in detail, including how to create an annotated dataset for its training and the formalization of the APR framework. Finally, we introduce the inference procedure of style transfer based on the LLM and our prompt router in our APR.

Style-Transfer Score Function A style-transfer score function related to evaluation metrics is commonly used for more directly accessing the model performance during training or selecting outputs during inference. We use a style-transfer score function for annotating candidate prompts and outputs reranking as in (Suzgun, Melas-Kyriazi, and Jurafsky 2022). Specifically, our score function has the same formulation with (Deng et al. 2022), which is defined as:

$$S(x^s, y^t) = Content(x^s, y^t) + Style(y^t), \qquad (1)$$

where $Content(\cdot)$ and $Style(\cdot)$ measure the content similarity between source text x^s and output y^t, and the target style strength of y^t, respectively. For the $Content(\cdot)$, we implement it with BERTScore (Zhang et al. 2020), a BERT-base metric; For the $Style(\cdot)$, we implement it with a style classifier trained on the sampled raw training text (100 samples per style) of the specific style-transfer task.

Prompt Annotation What we expect our prompt router to do is adaptively select a prompt q that is better than other prompts in the candidate set Q for the input sentence x^s. For

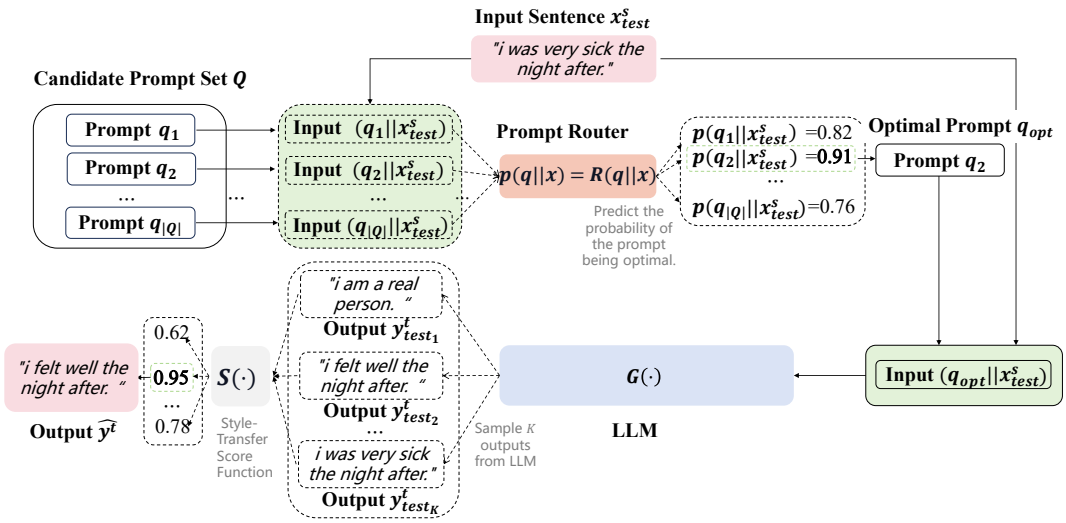

Figure 2: The overview of our APR framework for inference. Given an input sentence x^s_{test} with a negative sentiment. The prompt router $R(\cdot)$ of our APR predicts the optimal prompt q_{opt} from the task-specific prompt set $Q = \{q_m\}^{|Q|}_{m=1}$ by calculating the probability $p(q_m||x^s_{test})$ for each prompt q_m. This selected prompt, combined with the input sentence as $q_{opt}||x^s_{test}$, is then fed into the LLM $G(\cdot)$ for style transformation. To obtain high-quality output, we sample multiple potential outputs $y^t_{test\,j}|^K_{j=1}$ from the LLM and score them using a style-transfer score function $S(\cdot)$. The output $\hat{y^t}$ with the highest score, like *"i felt well the night after"*, is considered the final style-transfer result generated by the LLM.

this purpose, we train a prompt router in a supervised manner. To obtain the labeled training data, we annotate each candidate prompt for different input sentences.

First, we sample a bunch of source style sentences $X^s = \{x^s{}_1, ..., x^s_n, ...x^s_N\}$ unless otherwise specified) from the source style dataset D^s. For input sentence x^s_n, we assign a score independently to each prompt q_m in the candidate prompt set $Q = \{q_1, ..., q_m, ..., q_{|Q|=M}\}$. To score each prompt q_m given input x^s_n, we first add the input sentence x^s_n to the candidate prompt q_m to obtain the complete input $q_m||x^s_n$. Then, we feed input $q_m||x^s_n$ to the LLM $G(\cdot)$, which generates K candidate outputs y^t_k by sampling from its output distribution P:

$$P = G(q_m||x^s_n). \tag{2}$$

Next, we score each candidate output by the above style-transfer score function $S(\cdot)$ and finally assign a score $o_{m,n}$ to prompt q_m, where $o_{m,n}$ is the maximum style-transfer score of all sampled outputs:

$$o_{m,n} = max(S(x^s_n, y^t_1), ..., S(x^s_n, y^t_k), ..., S(x^s_n, y^t_K)). \tag{3}$$

After scoring all the prompts in the candidate set Q for the input x^s_n, we divide the set Q into a positive sample set Q_{pos} and a negative sample set Q_{neg}. Here, we use a dynamic selection strategy based on the statistical characteristics of the prompt scores. First, we calculate the mean u and standard deviation v of scores over candidate prompts for input x^s_n:

$$u = \frac{1}{M}\sum_{m=1}^{M} o_{m,n}, \tag{4}$$

$$v^2 = \frac{1}{M-1}\sum_{m=1}^{M}(o_{m,n} - u)^2. \tag{5}$$

Then, we apply $h = u + v$ as the threshold for sorting out prompts. The prompts whose scores are greater than h are chosen as positive samples, denoting as Q_{pos}, and the rest are as negative samples, denoting as Q_{neg}.

$$Q_{pos} = \{q_m | m = 1, 2, ..., M, o_{m,n} \geq h\}, \tag{6}$$

$$Q_{neg} = \{q_m | m = 1, 2, ..., M, o_{m,n} < h\}, \tag{7}$$

Finally, we assign a hard label $c \in C = \{1, 0\}$ to each prompt. For positive samples from set Q_{pos}, we set their labels $c = 1$; for negative samples from set Q_{neg}, we set their labels $c = 0$. We find that when the annotated prompt data is limited, it is easier for the router to learn hard labels rather than directly learning the computed floating point score. For every source sentence x^s_n within the sampled dataset X^s, we assign labels to each prompt q_m in the candidate prompt set Q according to the style-transfer score.

Formalization of APR Based on the annotated prompts data, we train our prompt router $R(\cdot)$, which consists of two parts: a BERT encoder $BERT(\cdot)$ and one fully connected (FC) layer $FFN(\cdot)$ followed by a sigmoid function $Sigmoid(\cdot)$. The prompt router $R(\cdot)$ takes the complete input $q_m||x^s_n$ as its encoding sequence and returns the probability p of the prompt q_m to be optimal for input text x^s_n. This can be formulated as:

$$p = R(q_m||x^s_n) = Sigmoid(FFN(BERT(q_m||x^s_n))) \tag{8}$$

In our experiments, we initialize $BERT(\cdot)$ using a pre-trained BERT-base model (Devlin et al. 2019) and take the output of the CLS token as the input vector of the FC layer.

The training instance includes the input sentence, prompt, and annotated label: $(q_m||x_n, c)$. To optimize the prompt router, we employ the Binary Cross-Entropy (BCE) loss as the training objective. Then, we run the prompt router on all the training instances based on the sampled data and compute the average loss. After training, for a given input text and a prompt from the candidate set, our prompt router can provide the optimal prompt.

Inference During testing, given an input sentence x^s_{test}, we adaptively identify the optimal prompt q_{opt} from the candidate set Q according to the prompt router (see Figure 2).

First, we enumerate each prompt q_m in the candidate set $Q = \{q_m\}_{i=1}^{|Q|=M}$ and create the router input by adding the input sentence x^s_{test} to each candidate prompt to form $q_m||x^s_{test}$ fed to prompt router $R(\cdot)$. The router predicts the probability $p(q_m||x^s_{test})$ of each prompt being optimal for the test input:

$$p(q_m||x^s_{test}) = R(q_m||x^s_{test}). \tag{9}$$

The prompt q_{opt} with the highest probability is regarded as the optimal prompt:

$$q_{opt} = \arg\max_{q_m} p(q_m||x^s_{test}), q_m \in Q. \tag{10}$$

Then, We select the optimal prompt q_{opt} along with input sentence x^s_{test} for the LLM to perform style transfer. To obtain the high-quality generated output from the LLM, we use the reranking method proposed by Suzgun, Melas-Kyriazi, and Jurafsky (2022). Specifically, We create the LLM input $q_{opt}||x^s_{test}$ by adding x^s_{test} to q_{opt} and feed it to the LLM $G(\cdot)$. We sample K candidate outputs $Y^t_{test} = \{y^t_{test_j}\}_{j=1}^K$ from output distribution P_{test} of the LLM:

$$P_{test} = G(q_{opt}||x^s_{test}), \tag{11}$$

and we score all these candidate outputs via Equation (1) and select the output $\hat{y^t}$ from Y^t_{test} with highest score:

$$\hat{y^t} = \arg\max_{y^t_{test_j}} S(x^s, y^t_{test_j}), y^t_{test_j} \in Y^t_{test}. \tag{12}$$

Experiments

Datasets We conduct experiments on 4 public TST datasets, including **Yelp**, **Amazon**, **GYAFC**, and **Shakespeare**. **Yelp** is a sentiment transfer dataset with positive and negative reviews (Zhang, Zhao, and LeCun 2015). **Amazon** is another sentiment transfer dataset including reviews from the Amazon website (Li et al. 2018). **GYAFC** is a sentence formality transfer dataset (Rao and Tetreault 2018). In our experiments, we select the *Family & Relationships* domain for a fair comparison with prior work. **Shakespeare** is a dataset for Shakespeare authorship transfer (Xu et al. 2012). Datasets statistics are shown in the supplementary file.

Evaluation Metrics We evaluate the performance of a TST model from three dimensions. The first is **Style Transfer Strength** (**Sty.**). It measures whether the generated sentence has the target style attribute. In our experiments, we fine-tune a classifier for each experimental dataset using the pre-trained Roberta-Large (Liu et al. 2019). The second is **Content Preservation** (**Con.**). It measures the content similarity of the generated output and source sentence. Following (Deng et al. 2022), we use a BERT-based(Zhang et al. 2020) metric to calculate the semantic similarity of the output and source sentence. The final is **Fluency** (**Flu.**), which measures whether the generated sentence is fluent. In our experiments, we use a trained grammatical classifier (Krishna, Wieting, and Iyyer 2020) to calculate fluency.

To evaluate the overall generation quality, we calculate the average of sentence-level joint score $J(\cdot)$ over the test set X^s_{test}, following the definition in (Krishna, Wieting, and Iyyer 2020):

$$J(Style, Content, Fluency) = \sum_{x^s_{test} \in X^s_{test}} \frac{Style(x^s_{test}) \cdot Content(x^s_{test}) \cdot Fluency(x^s_{test})}{|X^s_{test}|}. \tag{13}$$

We also report the popular geometric mean (GM) of scores in all three dimensions.

Baselines Fine-tuning and prompt-based baselines are as follows:

- **Fine-tuning(FT)**: Fine-tuning methods could serve as a strong baseline, as they use the full training data to update the LLM parameters. We choose DiRR(Liu, Neubig, and Wieting 2021) and STRAP(Krishna, Wieting, and Iyyer 2020) as our fine-tuning baseline. They fine-tune GPT-2-XL and GPT-2-Large, respectively.
- **Prompt-and-Rerank (P&R)** (Suzgun, Melas-Kyriazi, and Jurafsky 2022): P&R is the state-of-the-art prompt-based method for TST by reranking. They conducted a detailed discussion of specific syntax and semantics of the prompt and manually designed a fixed prompt template for arbitrary TST, formally similar to the previous study (Reif et al. 2022).
- **RLPrompt** (Deng et al. 2022): RLPrompt is a state-of-the-art discrete prompt search method. They use reinforcement learning to search optimal prompt tokens with the best performance on the validation set.
- **Low Perplexity (LP)** (Gonen et al. 2022): Gonen et al. (2022) recently proposed a method for automatic prompt selection, which chooses i (e.g., 3) lowest perplexity prompts in a human-readable prompt set.

Implementation Details We conduct experiments on three different scale LLMs: GPT-2-XL (Radford and Wu 2019) with 1.5B parameters, GPT-J-6B (Wang and Komatsuzaki 2021), and GPT-3.5-turbo with 175B parameters. For the fine-tuning baselines, we evaluate their released outputs using the same evaluation metric; for APR and other prompt-based methods, we conduct experiments under a few-shot setting (100 training examples per style). We train a few-shot classifier in the style-transfer scoring function with a

small number of samples, which is used for labeling prompts and reranking. We employ the same decoding and reranking strategy for both APR and other prompt-based baselines. Decoding has been done with top-10 sampling for GPT-2-XL and GPT-J-6B, while GPT-3.5 uses its default setting. We sample 30 candidate outputs for GPT-2-XL and GPT-J-6B, and 5 for GPT-3.5 due to cost considerations. For more details, please see the supplementary file.

Experimental Results

Main Results Table 1 shows the performance of our APR and baselines on different TST datasets and different scales LLMs. Compared to fine-tuning approaches, APR almost always generates outputs with better style-transfer quality on different style-transfer datasets, as measured by joint score and geometric mean, despite using significantly less training data (only 100-shot for each style). In addition, significantly higher fluency scores indicate that APR consistently generates more fluent and grammatical text than fine-tuning baselines. This is likely due to the inherent conflict between fine-tuning an LLM on style data and its pre-training task of predicting the next token, which can compromise the model's ability to generate coherent text. In general, APR outperforms previous fixed prompt-based approaches in terms of the joint score $J(\cdot)$ and geometric mean (GM) on different scale LLMs and TST datasets. Compared to P&R, adaptive prompts are significantly more effective in enhancing the performance of GPT-2-XL on both GYAFC and Shakespeare, leading to a remarkable 39% and 37% improvement in joint score, as well as 11% and 11% improvement in GM score. The formality and authorship transfer tasks tend to be more challenging than sentiment transfer, highlighting the need for adaptively-routed prompts for LLMs with limited size and ability to handle complex style-transfer tasks. and APR's performance is better and more stable than RLPrompt and LP. Additionally, APR provides richer style-transfer information to LLMS by routing different prompts, resulting in higher scores of style-transfer strength.

Human Evaluation To further verify the efficacy of APR, we also conduct human evaluation. Here, we mainly focus on the Amazon dataset, because APR doesn't demonstrate a significant advantage over other baselines when applied to Amazon using GPT-2-XL, as indicated by automated metrics. Following the procedure of (Suzgun, Melas-Kyriazi, and Jurafsky 2022), we select 50 source sentences from Amazon, with 25 positive and negative examples each, and corresponding outputs of DiRR(Liu, Neubig, and Wieting 2021), RLPrompt(Deng et al. 2022), P&R(Suzgun, Melas-Kyriazi, and Jurafsky 2022), and APR. For a clearer comparison, we include human references for each example. To ensure a fair evaluation, we recruited three graduate students who are fluent in English to rate the different outputs and human references based on a 1-5 Likert scale in terms of content preservation, target style strength, and fluency. The evaluation is conducted on the same 50 examples, with the order of the different outputs shuffled. The results in Tabel 3 demonstrate that our APR outperforms all baselines in terms of preserving content and fluency. Although

	Con.	Sty.	Flu.	J	GM	Con.	Sty.	Flu.	J	GM
Method	**Dataset:Yelp**					**Dataset:Amazon**				
	GPT-2									
DiRR (FT)	**76.6**	**94.0**	73.1	54.1	80.7	**77.5**	53.5	74.1	28.9	67.5
RLPrompt	67.3	76.7	91.0	44.2	77.7	59.5	47.6	**90.6**	23.6	63.5
P&R	72.9	82.7	89.4	53.8	81.4	61.5	**59.3**	84.4	**29.0**	67.5
LP	72.2	82.5	90.4	53.6	81.3	61.2	56.4	86.6	27.8	66.8
APR (Ours)	71.4	84.4	**91.1**	**54.8**	**81.9**	61.5	58.6	85.8	28.7	**67.6**
	GPT-J-6B									
P&R	74.7	**84.8**	90.1	56.5	82.9	64.6	55.6	87.0	28.2	67.8
LP	75.8	84.1	**90.8**	57.6	83.3	**66.9**	53.6	88.3	29.8	68.1
APR (Ours)	**78.6**	83.0	90.2	**59.1**	**83.8**	56.1	**68.0**	**89.6**	**31.4**	**69.9**
	GPT-3.5-turbo									
P&R	**69.7**	86.1	**96.0**	57.4	83.2	61.9	55.3	**95.8**	32.4	68.9
APR (Ours)	67.7	**94.7**	94.5	**60.3**	**84.6**	**62.5**	**58.9**	94.3	**33.9**	**70.3**
Method	**Dataset: GYAFC**					**Dataset: Shakespeare**				
	GPT-2									
DiRR (FT)	**78.5**	62.6	80.4	41.2	73.4	-	-	-	-	-
STRAP (FT)	-	-	-	-	-	40.7	**73.9**	87.8	26.4	64.2
RLPrompt	75.1	60.5	94.6	40.3	75.4	**57.8**	54.9	87.5	26.1	65.2
P&R	61.3	65.1	94.0	34.5	72.1	57.0	44.7	85.7	20.4	60.2
LP	65.1	71.8	95.4	42.0	76.3	55.8	57.8	87.8	26.4	65.6
APR (Ours)	67.2	**78.3**	**96.5**	**47.8**	**79.8**	55.5	60.3	**89.5**	**28.0**	**66.9**
	GPT-J-6B									
P&R	68.6	**81.3**	95.6	51.4	81.1	**61.8**	66.9	87.7	35.0	71.3
LP	71.7	74.1	96.0	49.2	79.8	61.0	66.0	88.6	34.7	70.9
APR (Ours)	**73.7**	79.6	**97.6**	**55.5**	**83.0**	61.2	**71.1**	**90.2**	**38.3**	**73.2**
	GPT-3.5-turbo									
P&R	50.5	**99.6**	**99.2**	50.0	79.3	50.3	**92.5**	**95.8**	**43.7**	**76.4**
APR (Ours)	**74.0**	76.8	98.6	**56.4**	**82.4**	**50.7**	91.0	95.3	43.6	76.0

Table 1: Comparison with baselines on different TST datasets and LLMs. $J(\cdot)$ and GM are our main metrics, which measure the overall style-transfer quality of generations. The best result on each aspect is highlighted in bold and the second best result is underscored.

having a slightly lower style transfer strength compared to P&R, our APR obtains the closest style-transfer quality to ground truth data, as evidenced by the geometric mean of the scores across all three dimensions.

Effectiveness Analysis of Adaptively Routing To analyze the effectiveness of the router in APR, we compare two other strategies of selecting prompts from the prompt set: 1) Select a prompt randomly (for each input); 2) Select a fixed prompt with the top style transfer score (STS) on the validation set (we take the average performance of top-3 score prompts). The result is shown in Table 4. We can observe that the performance of prompt selection based on style transfer score is significantly better than random selection. Compared with the fixed prompts selected according to the score, the prompt selected adaptively by the router achieves better style transfer performance. This shows that adaptively routing prompts by our prompt router is effective and necessary in our APR framework.

Analysis of Positive Prompt Selection We analyze the impact of different strategies for annotating positive prompts in APR on model performance. We compare two strategies: 1) Dynamic: Select positive prompts for each input accord-

System		Prompt
Source text	*My ex of 3 years cheated.*	
Ground Truth	*My three years ex cheated on me.*	
RLPrompt	*My ex of **3** years **cheated**.*	Iran Include shortened plural Replace *My ex of 3 years cheated.*
P&R	*My ex of **3** years **cheated, and I told him to go to counseling**.*	Here is a text, which is informal: {*My ex of 3 years cheated.*} Here is a rewrite of the text, which is more formal: {
APR	*My ex of three years cheated on me.*	User: Can you help me rephrase this informal text into a more formal tone? "*My ex of 3 years cheated.*" Assistant: Of course! Here's a more formal version: "
Source text	*If He Likes You, You'll Know!*	
Ground Truth	*If he is interested in you, you will just know.*	
RLPrompt	*If he likes you, you**'ll** know.*	Iran Include shortened plural Replace *If He Likes You, You'll Know!*
P&R	*If **He** likes you, you**'ll** know.*	Here is a text, which is informal: {*If He Likes You, You'll Know!*} Here is a rewrite of the text, which is more formal: {
APR	*If he likes you, you will know.*	You come across an informal sentence that needs to be rewritten formally: "*If He Likes You, You'll Know!*" How would you transform this sentence into a more formal tone? "

Table 2: Example outputs and their corresponding prompts from prompt-based methods on the informal to formal transfer task. Improper generations are highlighted in bold.

Setting	Con.	Sty.	Flu.	**GM**
Ground-Truth	4.91	3.81	4.87	4.50
DiRR	4.53	2.69	4.69	3.85
RLPrompt	3.99	2.55	4.73	3.63
P&R	4.45	**3.59**	4.61	4.19
APR	**4.65**	3.55	**4.77**	**4.29**

Table 3: Human Evaluation on Amazon on the 5-Likert score. The best result on each aspect is highlighted in bold and the second best result is underscored.

Setting	Con	Sty.	Flu.	**J**	**GM**
Random	59.1	72.9	96.2	38.3	74.5
STS	65.9	77.5	96.1	46.2	78.9
Router	**67.2**	**78.3**	**96.5**	**47.8**	**79.8**

Table 4: Comparing of prompt routing strategies in APR on GYAFC using GPT-2-XL for inference.

Setting	#Avg. Pos.	Con.	Sty.	Flu.	**J**	**GM**
Top-$i(i=1)$	166.7	**68**	71.9	90.5	43.8	76.1
Top-$i(i=2)$	329.5	66.4	75.1	91.8	45	76.9
Top-$i(i=3)$	475.8	67.7	73.5	91.2	44.9	76.8
Dynamic	329.0	67.4	**75.4**	**91.9**	**46.1**	**77.5**

Table 5: Comparison of positive prompts selection strategies: Average performance with GPT-J-6B on 4 datasets. #Avg. Pos. indicates the average number of positive prompts across all the prompt training data.

ing to the statistical characteristics of their scores; 2) Top-i: Select the top-i score prompts as the positive prompts. We count the average number of prompts labeled as positive in all the prompt training data built for different styles, and the average performance (we calculate the average score of each evaluation metric) on the 4 TST datasets with GPT-J-6B. The average scores of $J(\cdot)$ and GM in Table 5 demonstrate the dynamic strategy achieves the best performance.

Case Study Table 2 shows the prompts and outputs of our APR and the previous prompt-based methods on the informal to formal transfer task of the GYAFC dataset. The optimal prompt from RLPrompt consists of five different words "Iran Include shortened plural Replace". However, these words are challenging for humans to understand, and it's hard for us to intuitively understand how the LLM performs style transfer based on these messy words. Regarding performance, we observe that fixed prompts struggle with diverse sentences. Considering the informal sentence: *"If He Likes You, You'll Know!"*, RLPrompt and P&R successfully alter the tone from exclamatory to declarative and modify the rectify incorrect capitalization (P&R ignores the "*H*" in the second word "*He*"), although they ignore the informal abbreviations "*'ll*". However, for the sentence: *"My ex of 3 years cheated."*, RLPrompt directly copies it as output while P&R exhibits more uncontrollable by introducing expansions without making any modifications to the source sentence. In contrast, for each informal source text, our APR adaptively routes two prompts in different types, leading the LLM to reach superior style transfer performance.

Conclusion

In this paper, we propose a novel prompt-based framework for arbitrary text style transfer to effectively address the importance of selecting suitable prompts for different input sentences while ensuring that they remain readable to humans. Our APR framework, adaptively routing human-readable task-specific prompts, is both resource-efficient and effective during training. Besides, our APR can be seamlessly adapted to various sizes of LLMs. Through extensive experiments on various TST datasets and LLMs, we demonstrate that the APR framework achieves superior performance compared to existing prompt-based even fine-tuning methods. In summary, this work contributes to the advancement of TST research and highlights the importance of selecting appropriate prompts in achieving effective and efficient prompt-based TST systems.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2021ZD0111601, in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62276283 and No. 62206314, in part by Guangdong Basic and Applied Basic Research Foundation under Grant No. 2023A1515012985 and No. 2022A1515011835, in part by Basic and Applied Basic Research Special Projects under Grant SL2022A04J01685, and China Postdoctoral Science Foundation funded project under Grant No. 2021M703687. We would also like to thank Guangdong Province Key Laboratory of Information Security Technology.

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Deng, M.; Wang, J.; Hsieh, C.-P.; Wang, Y.; Guo, H.; Shu, T.; Song, M.; Xing, E.; and Hu, Z. 2022. RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 3369–3391. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Gao, T.; Fisch, A.; and Chen, D. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Gonen, H.; Iyer, S.; Blevins, T.; Smith, N. A.; and Zettlemoyer, L. 2022. Demystifying Prompts in Language Models via Perplexity Estimation. arXiv:2212.04037.

Gu, Y.; Han, X.; Liu, Z.; and Huang, M. 2021. PPT: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward Controlled Generation of Text. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1587–1596. PMLR.

Jiang, Z.; Xu, F. F.; Araki, J.; and Neubig, G. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8: 423–438.

John, V.; Mou, L.; Bahuleyan, H.; and Vechtomova, O. 2019. Disentangled Representation Learning for Non-Parallel Text Style Transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 424–434. Florence, Italy: Association for Computational Linguistics.

Kaushik, D.; Hovy, E.; and Lipton, Z. 2020. Learning The Difference That Makes A Difference With Counterfactually-Augmented Data. In *International Conference on Learning Representations*.

Krishna, K.; Wieting, J.; and Iyyer, M. 2020. Reformulating Unsupervised Style Transfer as Paraphrase Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 737–762. Online: Association for Computational Linguistics.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Li, J.; Jia, R.; He, H.; and Liang, P. 2018. Delete, Retrieve, Generate: a Simple Approach to Sentiment and Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1865–1874. New Orleans, Louisiana: Association for Computational Linguistics.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4582–4597. Online: Association for Computational Linguistics.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Liu, X.; Ji, K.; Fu, Y.; Tam, W.; Du, Z.; Yang, Z.; and Tang, J. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 61–68.

Liu, X.; Ji, K.; Fu, Y.; Tam, W. L.; Du, Z.; Yang, Z.; and Tang, J. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Liu, Y.; Neubig, G.; and Wieting, J. 2021. On Learning Text Style Transfer with Direct Rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4262–4273. Online: Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Lu, Y.; Bartolo, M.; Moore, A.; Riedel, S.; and Stenetorp, P. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Luo, F.; Li, P.; Zhou, J.; Yang, P.; Chang, B.; Sui, Z.; and Sun, X. 2019. A dual reinforcement learning framework for unsupervised text style transfer. *arXiv preprint arXiv:1905.10060*.

Luo, G.; Han, Y. T.; Mou, L.; and Firdaus, M. 2023. Prompt-Based Editing for Text Style Transfer. *arXiv preprint arXiv:2301.11997*.

Mishra, S.; Khashabi, D.; Baral, C.; Choi, Y.; and Hajishirzi, H. 2022. Reframing Instructional Prompts to GPTk's Language. In *Findings of the Association for Computational Linguistics: ACL 2022*, 589–612. Dublin, Ireland: Association for Computational Linguistics.

Nogueira dos Santos, C.; Melnyk, I.; and Padhi, I. 2018. Fighting Offensive Language on Social Media with Unsupervised Text Style Transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 189–194. Melbourne, Australia: Association for Computational Linguistics.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; and Miller, A. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2463–2473. Hong Kong, China: Association for Computational Linguistics.

Radford, A.; and Wu, J. 2019. Rewon child, david luan, dario amodei, and ilya sutskever. 2019. *Language models are unsupervised multitask learners. OpenAI blog*, 1(8): 9.

Rao, S.; and Tetreault, J. 2018. Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 129–140. New Orleans, Louisiana: Association for Computational Linguistics.

Reif, E.; Ippolito, D.; Yuan, A.; Coenen, A.; Callison-Burch, C.; and Wei, J. 2022. A Recipe for Arbitrary Text Style Transfer with Large Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 837–848. Dublin, Ireland: Association for Computational Linguistics.

Sanh, V.; Webson, A.; Raffel, C.; Bach, S. H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Scao, T. L.; Raja, A.; et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Schick, T.; and Schütze, H. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 255–269. Online: Association for Computational Linguistics.

Shen, T.; Lei, T.; Barzilay, R.; and Jaakkola, T. 2017. Style transfer from non-parallel text by cross-alignment. *Advances in neural information processing systems*, 30.

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4222–4235.

Suzgun, M.; Melas-Kyriazi, L.; and Jurafsky, D. 2022. Prompt-and-Rerank: A Method for Zero-Shot and Few-Shot Arbitrary Textual Style Transfer with Small Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2195–2222. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Wang, B.; and Komatsuzaki, A. 2021. GPT-J-6B: A 6 billion parameter autoregressive language model.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Xu, W.; Ritter, A.; Dolan, B.; Grishman, R.; and Cherry, C. 2012. Paraphrasing for Style. In *Proceedings of COLING 2012*, 2899–2914. Mumbai, India: The COLING 2012 Organizing Committee.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zhao, Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, 12697–12706. PMLR.

Zheng, Y.; Chen, Z.; Zhang, R.; Huang, S.; Mao, X.; and Huang, M. 2021. Stylized Dialogue Response Generation Using Stylized Unpaired Texts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16): 14558–14567.

Zhu, K.; Wang, J.; Zhou, J.; Wang, Z.; Chen, H.; Wang, Y.; Yang, L.; Ye, W.; Gong, N. Z.; Zhang, Y.; et al. 2023. PromptBench: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. *arXiv preprint arXiv:2306.04528*.