Step-by-Step: Controlling Arbitrary Style in Text with Large Language Models

Pusheng Liu[1,2,3], **Lianwei Wu**[1,2,3]*, **Linyong Wang**[1,2,3], **Sensen Guo**[4], **Yang Liu**[5]
[1]National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, School of Computer Science, Northwestern Polytechnical University, China
[2]Research & Development Institute of Northwestern Polytechnical University in Shenzhen, China
[3]Chongqing Science & Technology Innovation Center of Northwestern Polytechnical University, China
[4]State Key Laboratory of Integrated Service Networks, Xidian University, China
[5]School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University, China
lps@mail.nwpu.edu.cn, wlw@nwpu.edu.cn, guosensen@xidian.edu.cn
{linyongw5, yangliuyh}@gmail.com

Abstract

Recently, the autoregressive framework based on large language models (LLMs) has achieved excellent performance in controlling the generated text to adhere to the required style. These methods guide LLMs through prompt learning to generate target text in an autoregressive manner. However, this manner possesses lower controllability and suffers from the challenge of accumulating errors, where early prediction inaccuracies might influence subsequent word generation. Furthermore, existing prompt-based methods overlook specific region editing, resulting in a deficiency of localized control over input text. To overcome these challenges, we propose a novel three-stage prompt-based approach for specific region editing. To alleviate the issue of accumulating errors, we transform the text style transfer task into a text infilling task, guiding the LLMs to modify only a small portion of text within the editing region to achieve style transfer, thus reducing the number of autoregressive iterations. To achieve an effective specific editing region, we adopt both prompt-based and word frequency-based strategies for region selection, subsequently employing a discriminator to validate the efficacy of the selected region. Experiments conducted on several publicly competitive datasets for text style transfer task confirm that our proposed approach achieves state-of-the-art performance.

Keywords: text style transfer, natural language generation, large language models

1. Introduction

Text style transfer (TST) is an important task in natural language generation, aiming to change the style of text (e.g., emotion, politeness, formality) while preserving its content semantics information (Jin et al., 2022; Hu et al., 2022). Currently, TST has been widely applied in various domains, including sentiment transfer (Hu et al., 2017; Shen et al., 2017), improving online community environments (Moskovskiy et al., 2022; Liu et al., 2021), personal privacy protection (Prabhumoye et al., 2018; Lample et al., 2019), writing assistance (Liu et al., 2022), and data augmentation (Chen et al., 2022), attracting extensive attention from both the academic and industrial communities.

Previous work mainly focused on training a style transfer model to implement TST based on a large number of parallel or non-parallel corpora. However, they face the following challenges: 1) acquiring a sufficient amount of high-quality data is difficult; 2) a single model is limited to transforming text between a predefined set of styles and cannot achieve arbitrary style transfer. Recently, large language models (LLMs) have achieved remarkable performance in various NLP tasks and demonstrated astonishing text generation capabilities, enabling us to perform natural language generation tasks with zero-shot or few-shot settings. Inspired by this, researchers have explored prompt-based TST methods (Reif et al., 2022; Suzgun et al., 2022; Luo et al., 2023), querying LLMs using prompts like: "Here is a text: {your input statement.}. Here is a rewrite of the text, which is more positive: {". Subsequently, the LLM generates the complete target-style sentence in an autoregressive manner. These methods leverage LLMs to transfer the source style to any desired target style specified by

Text Style Transfer	Source Style	Target Style
Negative→Positive	This is a terrible restaurant.	This is a great restaurant.
Informal→Formal	U seem like a successful person.	You seem like a successful person.
Impolite→Polite	Delete the page and shut up.	Please delete the page.
Factual→Romantic	Two dogs play by a tree.	Two dogs in love are playing by a tree.
Shakespearean English →Modern English	To be, or not to be, that is the question.	To exist, or not to exist, that is the question.
Biased→Neutral	Go is the deepest game.	Go is one of the deepest game.
Offensive→Non-off	What the f*ck is your problem?	What is your problem?

Figure 1: Traditional methods can only handle 1-2 tasks, our method can achieve any text style transfer, showing 7 representative style transfer tasks.

* Corresponding author

LREC-COLING 2024, pages 15285–15295

the user, without the need for additional training or fine-tuning, thereby eliminating the requirement for training data and labels.

However, this autoregressive approach of generating complete sentences has the following limitations: **1)** Low controllability and error accumulation problem (Suzgun et al., 2022). Generating complete sentences requires the LLM to generate words one by one, which requires multiple forward passes. However, multiple forward passes can lead to error accumulation, where prediction errors in the early words of LLM can affect the predictions of later words, resulting in overall unsatisfactory model performance. **2)** Ignoring specific region editing and lacking control over specific semantic areas in the given text. **3)** Less reliable than trained methods (Reif et al., 2022). Compared to style transfer methods that are trained or fine-tuned on TST data, prompt-based methods exhibit lower stability and are more prone to generating output unrelated to the content.

To overcome the challenges, we propose **P**rompt-based **E**diting and **G**lobal **F**illing model (**PEGF**) for arbitrary style transfer. Especially, arbitrary style transfer refers to the fact that our model can achieve multiple styles of transfer, as shown in Figure 1, which implements 7 styles of transfer tasks. Inspired by the idea of chain-of-thought in LLM (Wei et al., 2022) and prototype editing methods (Li et al., 2018; Wang et al., 2022), we divided the model into three stages, namely the editing area acquisition stage, the validity verification stage of masked sequence, and the style information filling stage.

In detail, **1)** To address the issue of error accumulation caused by multiple forward passes of the LLM, we propose controlling the LLM to transition from generating complete sentences to modifying only a small number of words within specific regions, thereby reducing the number of inferences performed by the LLM. **2)** To obtain the editing region of the input text and guide the model to modify words within that region. In the editing region acquisition stage, the model identifies style words and masks them, resulting in a masked sequence. In the masked sequence validity verification stage, the model performs validity verification on the masked sequence. In the style information filling stage, the model generates the final output based on the valid masked sequence and the user-specified target style. **3)** To enhance the reliability, stability, and content preservation of the model and prevent it from generating outputs unrelated to the original input sentence content, we adopt an approach of implicitly marking stylistic words instead of directly deleting or replacing them with [MASK].

Extensive experiments confirm the superiority of PEGF. We summarize the main contributions and insights of this paper as follows:

- We propose a novel prompt-based editing and global filling method, which innovatively converts TST task into filling task to cope with the accumulation of errors in large models. It guides the LLM to edit text within specific semantic regions, enhancing its controllability, stability, and interpretability. To the best of our knowledge, we are the first to utilize prompt guidance for LLM to perform editing within specific semantic regions to achieve TST.
- To enhance content preservation, we design an implicit masking module that does not employ the traditional form of directly deleting the style words recognized by the model, but instead devise implicit masking to allow the model to retain more contextual information.
- Our experiments demonstrate that PEGF significantly outperforms state-of-the-art baselines on three publicly available benchmark datasets.

2. Related Work

TST is an important branch of natural language generation. Researchers have investigated the recent advancements in this task (Jin et al., 2022; Hu et al., 2022). Based on whether the model can achieve arbitrary TST and whether training data is required during the implementation of TST, research methods can be categorized into two types: traditional deep learning-based TST methods and prompt-based TST methods.

Traditional deep learning-based TST methods. Based on whether the process of style transfer separates the content and style of sentences, the mainstream methods can be broadly divided into the following two categories. **1)** Style-content disentanglement. In the process of achieving TST, the methods of separating the content and style of sentences are employed. Subsequently, the output is generated based on the sentence content and target style. These methods include the following two types. **a)** Implicit style-content disentanglement. Based on the adversarial network, learn the latent representation of content and style in sentences, and subsequently manipulate the latent representation directly to control the generation of text with specific styles (Shen et al., 2017). **b)** Explicit style-content disentanglement. By identifying and removing stylistic words in the sentences, followed by retrieving similar content in the target style, the final output is obtained through the decoder (Li et al., 2018). **2)** Without style-content disentanglement. This approach considers it challenging to separate the content and style within sentences. Therefore, it directly optimizes the mapping function between the input and output (Lample et al., 2019).

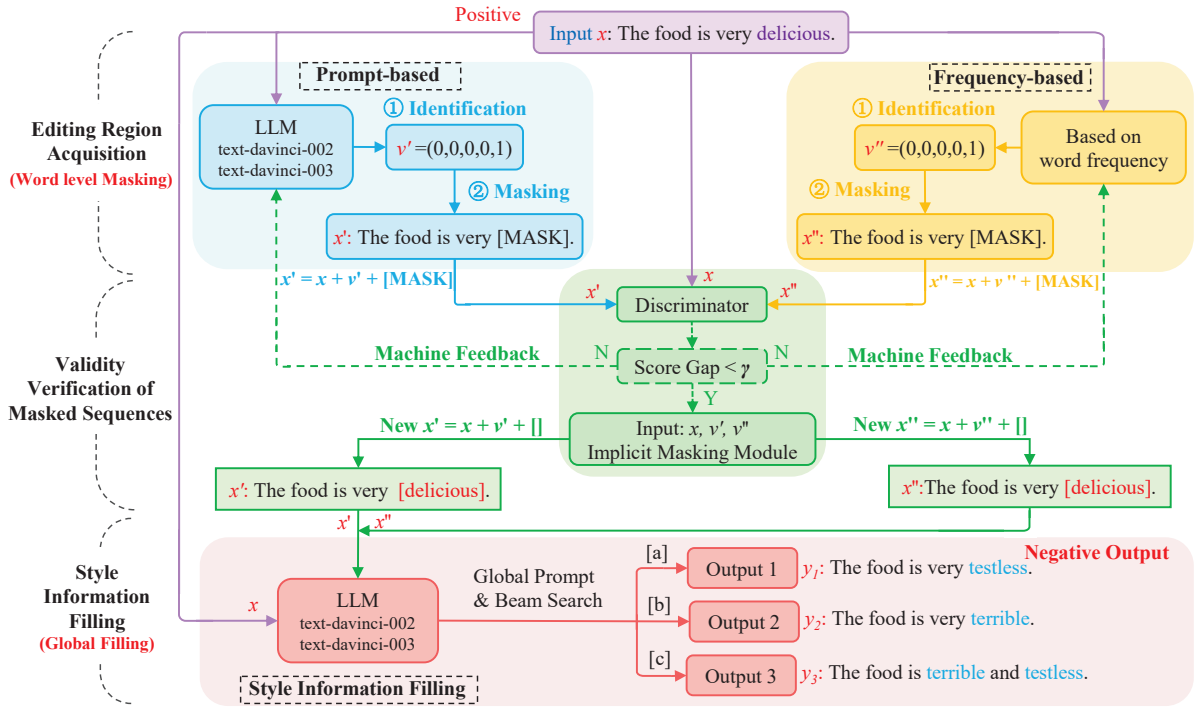

Figure 2: Our proposed PEGF model.

However, there are several limitations associated with these approaches: 1) The requirement of a substantial amount of parallel or non-parallel corpora poses a challenge as acquiring relevant data is difficult. 2) A single model can only achieve style transfer between a predefined set of styles and cannot accomplish arbitrary text style transfer.

Prompt-based TST methods. In recent years, the continuous development of LLM has helped significantly improve tasks in various NLP fields. Inspired by this, recent work in the TST field has found that TST can be achieved by leveraging LLM through prompt learning (Reif et al., 2022; Suzgun et al., 2022). This approach eliminates the need for training data and enables arbitrary TST. Reif et al. (2022) used prompt queries to generate sentences in different styles using LLM. Suzgun et al. (2022) queried LLM with multiple prompts to obtain multiple outputs, which were then scored and ranked, ultimately selecting the candidate sentence with the highest score as the output.

Our approach is also based on the prompt learning and achieves TST without the need for training data. However, unlike previous work, we guide the LLM to edit only a small amount of text within a specific region to accomplish TST. We transform the TST task into a three-stage text filling task, compared to the methods that generate complete sentences using autoregressive generation, our approach reduces the number of autoregressive steps, reduces the error accumulation of the LLM, and improves its controllability, interpretability, and stability. In particular, instead of directly deleting identified style words, we implicitly mask them, which preserves more content information.

3. Method

We propose a novel prompt-based PEGF model, which aims to control LLM to modify a small number of words in the local area of input text to improve the model performance of TST task. Inspired by chain-of-thought of LLM (Wei et al., 2022), we split the TST task into three stages, as shown in Figure 2, PEGF mainly consists of the editing region acquisition stage, the validity verification stage of masked sequences, and the style information filling stage. It's worth noting that the validity verification stage of the masked sequence can effectively enhance the accuracy of the model.

The roles of the three stages in the model are as follows: **1) Editing area acquisition stage.** This stage is used to acquire the editing area of the input text. In this stage, the model identifies style words and masks them to obtain a masked sequence. **2) Validity verification stage of the masked sequence.** This stage performs validity verification on the masked sequence obtained from the previous stage to ensure the correct acquisition of the editing regions. **3) Style information filling stage.** This stage generates the final output based on the valid masked sequence and the target style specified by the user. Next, we will describe each module of the model in detail.

3.1. Task Definition

Text style transfer aims to change the style of a text while preserving its content semantics. Given a set of styles $S = \{s, t\}$, where s and t represent two different styles, and two non-parallel corpora $D_s = \{X_i\}$ and $D_t = \{Y_j\}$, among them, $X_i = \{x_1, x_2, x_3...x_n\}$, $Y_j = \{y_1, y_2, y_3...y_n\}$, each x_i or y_j represents a sample in the corpus. The objective of TST task is to train a style transfer model that is capable of transferring the input text from the source style to the target style, while retaining the original content of the input text. In other words, it aims to convert a text x_i with style s to have style t while preserving its content semantics, or convert a text y_j with style t to have style s while preserving its content semantics. Formally, the TST can be expressed as follows:

$$y_i = f(x_i, s, t) \tag{1}$$

3.2. Editing Area Acquisition Stage

During this stage, the input text's editing area is obtained. Style words are identified and masked, with the masked region representing the editing area. We propose a two-step editing area acquisition strategy, which involves identification and masking. Style information is identified using two approaches: prompt-based and word frequency-based. Next, the identified style information is masked to generate the masked sequences x' and x''.

Specifically, the model takes the input text x and the source style s as inputs. The model identifies style words in the input text x and masks them. **1) Style word identification:** The model identifies style words by employing prompt-based and word frequency-based approaches. Subsequently, it generates a mask vector $v = [v_1, v_2, v_3, \ldots, v_{n-2}, v_{n-1}, v_n]$. It is noteworthy that mask vectors v' and v'' are generated separately based on the prompt and word frequency. **2) Masking style words:** For the mask vector v, where $v_i \in \{0, 1\}$, it indicates whether the $i-th$ word in the input text x is a style word. If v_i = 0, it means that the $i-th$ word in the source text contains less style information and is more related to content, thus it is a content word and should be preserved. Conversely, if $v_i = 1$, it means that the $i-th$ word in the input text x contains more style information, indicating it is a style word, and should be masked.

For example, assuming the input sentence x is "*The waiters in this restaurant are very **polite** and the food is **delicious***." In this stage, the model first identifies the style words and generates a mask vector $v = [0, 0, 0, 0, 0, 0, 0, \mathbf{1}, 0, 0, 0, 0, 0, \mathbf{1}]$. This indicates that the model has recognized the style words "*polite*" and "*delicious*". Then, the words corresponding to the values of 1 in the vector are masked using special tokens, i.e., "*polite*" and "*delicious*" are masked. For instance, mask the style words with [MASK], resulting in the masked sequence: "*The waiters in this restaurant are very **[MASK]** and the food is **[MASK]***.".

Prompt-based edit region acquisition. Following prior studies of TST (Reif et al., 2022; Suzgun et al., 2022), in our experiments, we utilize two LLM models as our benchmark models: text-davinci-002 and text-davinci-003. To identify stylistic words, we design the following prompt templates:

This is a $[S]$ sentence: $\{x\}$, what is the $[S]$ score for each word? Assign scores within the range of -1 (very [s]) to+1 (very [t]).

where x represents an input sentence, S denotes a set of styles, and s and t are optional choices from S, representing the source style and the target style, respectively. Words exceeding a specified threshold score are designated as style words.

Word frequency-based edit region acquisition. Research has found that prompt-based methods have lower reliability and stability compared to methods based on data training or fine-tuning (Reif et al., 2022). To address this issue, we were inspired by prototype editing methods (Li et al., 2018). In the task of sentiment transfer, we additionally adopt a word frequency-based approach to identify style words, combining it with the prompt-based approach to enhance the stability and reliability of the model in acquiring edit regions.

Formally, for any given word w, we determine its status as a stylized word using the following equation:

$$f(w, s) = \frac{count(w, D_s) + \lambda}{(E_{s \in S, s \neq t} count(w, D_t)) + \lambda} \tag{2}$$

where w represents a word, λ is a smoothing parameter, $count(w, D_s)$ represents the number of occurrences of $n-gram(w)$ in D_s, and $count(w, D_t)$ represents the number of occurrences of $n-gram(w)$ in D_t. When $f(w, s)$ exceeds a specified threshold γ, we define w as a style word.

3.3. Validity Verification of Masked Sequences

Recently, it has been discovered that LLM can be guided in the right direction through human feedback, enabling the acquisition of desired answers (OpenAI, 2023). Inspired by this idea, in order to effectively guide LLM in recognizing style information and masking it, ensuring the effectiveness of the masked sequence obtained during the editing region acquisition phase, we adopt the concept of machine feedback and design a discriminator module. Furthermore, we design an implicit masking module to preserve more original content from the input sentences.

Discriminator module. To validate the effectiveness of the editing regions and ensure that style information is correctly identified and masked, we design a discriminator module. The discriminator determines the validity of the model's output masked sequences, denoted as x' and x''. The results are then fed back to the LLM and the frequency-based editing region acquisition module. It should be noted that the masked sequences entering the discriminator adopt the [MASK] masking scheme, where style words are replaced with [MASK].

Specifically, we utilize our well-trained classifier as our discriminator module, which takes as input the original text x and the masked sequences x' and x'' obtained by the editing region acquisition phase. Upon receiving these inputs, the discriminator scores each of the three input sequences individually, resulting in x_score, x'_score, and x''_score. Subsequently, the score differences between x' and x, as well as between x'' and x, are computed. If the difference exceeds a predefined threshold, it indicates the effectiveness of the obtained editing region, i.e., the masked sequence is valid. When the masked sequence is valid, it proceeds to the implicit masking module.

Implicit masking module. Previous work has employed direct deletion of style words or explicit replacement with [MASK] to obtain masked sequences, followed by direct style information filling in the next stage. However, we observe that style words also contain content information. For example, in the aforementioned example, the words "*polite*" and "*delicious*" contain strong style information but also imply the main content of the sentence, related to the waiter and food respectively. Directly deleting style words or replacing "*polite*" and "*delicious*" with [MASK] may cause loss of content information in the sentence, affecting the final performance of the model. Therefore, we implicitly mark the corresponding words with a value of 1 in the masking vector v using the delimiter "[]". In the aforementioned example, $v = [0, 0, 0, 0, 0, 0, 0, \mathbf{1}, 0, 0, 0, 0, \mathbf{1}]$, the corresponding words "*polite*" and "*delicious*" with a value of 1 are marked with the delimiter "[]" to obtain the masked sequence, i.e., "*The waiters in this restaurant are very **[polite]** and the food is **[delicious]**.*". Finally, "*[polite]*" and "*[delicious]*" represent the editing regions obtained by the model.

3.4. Styled Filling Stage

After receiving the editing area of the input text, the subsequent stage involves operations such as adding, deleting, modifying, and replacing words within the editable area to achieve text style transfer. In this paper, we propose a global prompt-based approach for style information filling, which takes into account the contextual content for style transfer.

During the style filling stage, there are three inputs: the source input text x, the masked sequence x' obtained based on the prompt, and the masked sequence x'' obtained based on word frequency. These three inputs are individually queried to the LLM using prompts, resulting in multiple candidate outputs. For prompt configuration, in order to adapt to our three-stage framework and align with the implicit masking module, we modify the prompt template by replacing "[MASK]" with specific words. We design the following three manually written template formats:

(**a**) Filling based on implicit masking: "Here is a text, which is: $\{x(s)\}$, Here is a rewrite of the text, replace $\{w\}$ makes the text more $[t]$: {".

(**b**) Filling based on explicit masking: "Here is a text, which is: $\{x(s)\}$, Here is a rewrite of the text, replace [MASK] makes the text more $[t]$: {".

(**c**) Vanilla: "Here is a text: $\{x(s)\}$, Here is a rewrite of the text, which is more $[t]$: {".

where $x(s)$ denotes the input text x with the source style s, while t represents the target style and w represents the style words in the input text x.

After obtaining multiple candidate outputs, we evaluate them based on accuracy, content preservation, and fluency. Finally, we select the candidate sentence with the highest score as the final output.

4. Experiments

4.1. Datasets

To validate the superiority of PEGF, we conduct experiments on a total of three competitive datasets, namely Yelp, Amazon, and GYAFC. For other types of style transfer executed by our model, please refer to Figure 1 and Figure 3.

Yelp: The Yelp dataset is collected from Yelp website, the largest business review website in the United States. The dataset consists of textual reviews, with each review labeled as either positive or negative (Zhang et al., 2015).

Amazon: Similar to the Yelp dataset, the Amazon dataset is composed of user reviews from the Amazon shopping website, wherein each review is labeled as positive or negative (Li et al., 2018).

GYAFC: The GYAFC dataset is a parallel corpus of informal and formal sentences. Each statement is labeled with formal or informal labels (Rao and Tetreault, 2018).

4.2. Evaluation Metrics

In order to ensure a fair comparison with previous research, we follow the evaluation methodology employed in prior studies (Reif et al., 2022) and adopt the following automatic evaluation metrics:

Style transfer accuracy (Acc). Measuring the conformance of output statements to the target

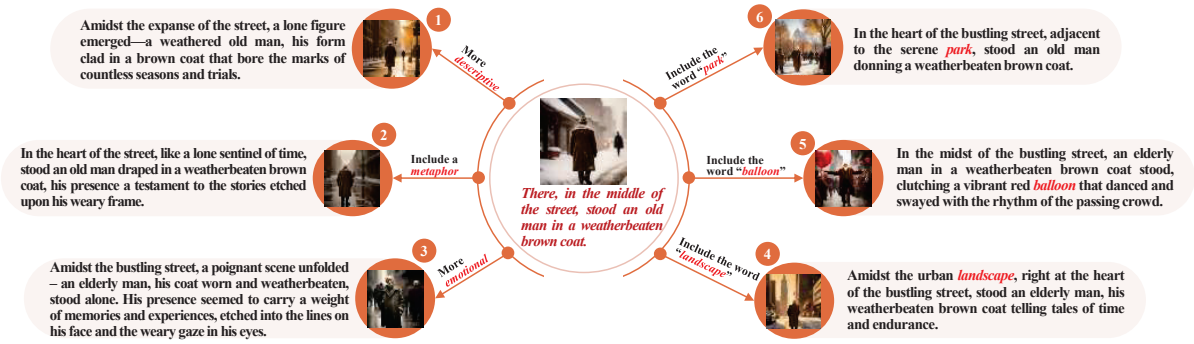

Figure 3: The multiple style transformation examples executed by our model.

style. We train a classifier on the corpus of interest and utilize it to estimate the proportion of generated outputs that match the desired style.

Content preservation (BLEU). Following the approach of Suzgun et al. (2022), we employ sBLEU (Post, 2018) to compute the reference-sBLEU(r-sBLEU) and self-sBLEU(s-sBLEU) scores.

Perplexity (PPL). The perplexity score measures an approximate value of the grammatical correctness of the candidate sentence y. It is worth noting that a higher PPL value indicates poorer model performance. We employ GPT-2-Large to assess the perplexity score of candidate sentences.

4.3. Implementation Details

Model choices. We employed two LLMs in our study, namely text-davinci-002 and text-davinci-003. We highlight that none of these models were fine-tuned or prompt-tuned.

Discriminator. The discriminator is designed to validate the effectiveness of the masked sequences x' and x''. The effectiveness is determined when x' and x'' satisfy the following conditions: 1) The score difference falls between 0.3-1, as the presence of a score difference indicates correct masking of style information. 2) The discriminator assigns a neutral score to the masked sequences, transitioning from the original input's positive or negative sentiment to neutral, indicating correct masking of style information. 3) To prevent the model from masking all vocabulary, the number of masked words is set to be no more than half of the input sentence. 4) To balance performance and computational costs and prevent the model from entering a loop, the maximum number of iterations is set to 5.

Sample quantity setting. Including zero-shot and few-shot settings. Zero-shot refers to the absence of reference examples during prompts. Following Reif et al. (2022) and Suzgun et al. (2022), we set the number of reference examples in the few-shot setting to 4.

4.4. Baselines

Due to the utilization of prompts in our approach, the training process is not required. Therefore, we primarily compared our method with the following three state-of-the-art prompt-based methods. In addition, we also compared some of the classical supervised methods in the field of TST.

Prompt-based methods: LLM$_{Aug}$ (Reif et al., 2022) enables direct acquisition of stylized sentences through prompting. **P&R** (Suzgun et al., 2022) achieves the generation of multiple outputs by utilizing various prompts and ranks and sorts them using a customized scoring function. **PB-E** (Luo et al., 2023) transforms the generation task into a classification task, leveraging a pre-trained language model for style classification and utilizing classification probabilities to calculate style scores.

Supervised methods: CrossAlignment (Shen et al., 2017) leverages refined alignment of latent representations to perform style transfer. **BackTrans** (Prabhumoye et al., 2018) uses back-translation to rephrase sentences and generate content from a latent representation. **MultiDecoder** (Fu et al., 2018) captures separate content representations and style features through adversarial networks for TST. **DeleteOnly** and **DeleteRetrieve** (Li et al., 2018) remove style words, find new words for the target style, and use a neural model to generate fluent, content-preserving output. **UnpairedRL's** (Xu et al., 2018) key idea is to build supervised training pairs by reconstructing the original sentence. **DualRL** (Luo et al., 2019) uses a dual reinforcement learning framework to directly transfer the style of the text via a one-step mapping model, without any separation of content and style. **ST-Multi-Class** and **ST-Conditional** (Dai et al., 2019) make no assumption about the latent representation of source sentence and equips the power of attention mechanism in Transformer. **B-GST** (Sudhakar et al., 2019) deletes style attributes from the source sentence by exploiting the inner workings of the Transformer.

Model	Yelp			
	Acc	r-sBLEU	s-sBLEU	PPL
Supervised				
CrossAlignment	0.73	7.8	18.3	217
BackTrans	**0.94**	2.0	46.5	158
MultiDecoder	0.49	13.0	39.4	373
DeleteOnly	0.84	13.4	33.9	182
DeleteRetrieve	0.91	14.7	36.4	180
UnpairedRL	0.49	16.8	45.7	385
DualRL	0.89	25.9	58.9	**133**
ST-Multi-Class	0.85	**26.4**	**63.0**	175
ST-Conditional	0.91	22.9	52.8	223
B-GST	0.83	21.6	46.5	158
Zero-shot or Few-shot (Prompt-based)				
LLM$_{Aug-0S}$	0.92	5.9	10.1	32
LLM$_{Aug-4S}$	0.81	23.5	**46.6**	79
P&R	0.88	23.0	45.9	80
PB-E	0.89	20.5	34.9	94
Our PEGF $_{0S}$	**0.94**	8.7	12.8	**29**
Our PEGF $_{4S}$	0.92	**25.3**	45.5	73

Table 1: Comparison of our method with previous works on Yelp dataset.

Dataset	Model	Acc	r-sBLEU	s-sBLEU	PPL
Amazon	P&R	0.65	21.5	31.4	**70**
	PB-E	0.76	33.2	44.7	98
	Our PEGF	**0.80**	**35.2**	**49.8**	86
GYAFC	P&R	0.85	36.4	49.6	68
	PB-E	0.81	37.7	**50.2**	87
	Our PEGF	**0.88**	**38.2**	46.4	**31**

Table 2: Comparison of our approach with previous prompt-based methods on the Amazon and GYAFC datasets.

Dataset	Model	Acc	r-sBLEU	s-sBLEU	PPL
Yelp	text-davinci-002	0.83	29.9	55.9	109
N→P	text-davinci-003	0.92	25.3	45.5	73
Amazon	text-davinci-002	0.77	37.8	50.2	82
N→P	text-davinci-003	0.80	35.2	49.5	86
GYAFC	text-davinci-002	0.85	35.39	43.63	34
Inf→F	text-davinci-003	0.88	38.2	46.40	31

Table 3: Results of the text-davinci-002 and text-davinci-003 models on different datasets. In which, N→P represents transforming negative style into positive style, while Inf→F indicates transforming informal style into formal style.

4.5. Main Results

Table 1 presents a comparison between our results on the Yelp dataset and the results of previous research methods. Despite not undergoing any training or fine-tuning, our approach is competitive compared to models specifically designed and trained for these tasks. Specifically, our model consistently generates smoother output compared to supervised methods, as measured by perplexity. In comparison to the recently proposed prompt-based methods, we follow the approach of Suzgun et al. (2022) and conduct experiments in a zero-shot and four-shot setting. Our proposed method achieves state-of-the-art performance in three aspects other than s-sBLEU. This is due to our proposed three-stage TST strategy, which avoids unnecessary edits, preserves more input content and ensures the validity of intermediate outputs.

Table 2 presents the results of our comparison with different prompt-based methods on the Amazon and GYAFC datasets (Suzgun et al., 2022; Luo et al., 2023). To ensure a fair comparison with previous prompt-based methods, we followed the approach of Suzgun et al. (2022) and conducted experiments in a four-shot setting. The results demonstrate that our approach exhibits slightly inferior performance in terms of PPL on the Amazon and in terms of s-sBLEU on the GYAFC. However, it achieves state-of-the-art performance in terms of ACC and r-sBLEU on both datasets.

Table 3 summarizes the results obtained using the text-davinci-002 and text-davinci-003 models on the Yelp, Amazon, and GYAFC datasets. Specifically, we observed that both models achieved satisfactory performance in TST. The text-davinci-002 model demonstrated better BLEU, while the text-davinci-003 model exhibited outstanding performance in terms of Acc and PPL.

4.6. Human Evaluation

To comprehensively evaluate the proposed PEGF model, we conducted a human evaluation to validate the effectiveness of the model. We randomly sampled 50 instances from the test set, and five evaluators provided ratings on a scale of 1 to 4. Figure 4 presents the human evaluation results of our method compared to three prompt-based approaches on the Yelp dataset. We observed that the PEGF model demonstrates excellent performance in terms of accuracy and content retention, while the performance of the four models is relatively similar in terms of fluency.

4.7. Detail Analysis

In this section, we conduct an in-depth analysis to evaluate the effectiveness of the PEGF model. Due to the constraints of time and resources, we choose the Yelp and Amazon datasets as our testing data.

Ablation study. To evaluate the contributions of key components in our model, we ablate PEGF into several modules.

w/o Discriminator: Indicates the ablation of the

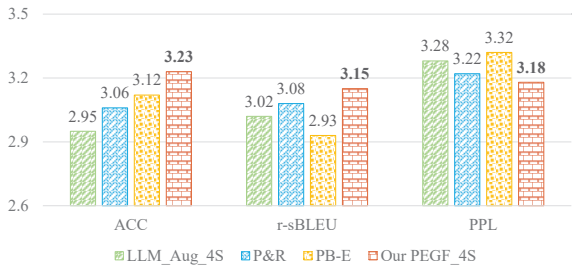

Figure 4: Human evaluation on the Yelp dataset.

Dataset	Model	Acc	BLEU	PPL
Yelp	PEGF	92	25.3	73
	w/o Discriminator	89	23.2	68
	w/o Implicit masking	86	21.6	65
	w/o Frequency-based	91	24.3	77
Amazon	PEGF	80	35.2	86
	w/o Discriminator	76	32.9	78
	w/o Implicit masking	73	33.7	76
	w/o Frequency-based	79	34.3	92

Table 4: Ablation study of the model on Yelp and Amazon datasets in the four-shot setting.

discriminator. Ablating this implies that the validity of the masked sequences, x' and x'', is not verified, while the other modules remain unchanged.

w/o Implicit masking: Represents the ablation of the implicit masking module. Removing this module results in the style words being directly replaced with [MASK] to obtain the masked sequence, while the other modules remain unchanged.

w/o Frequency-based: Signifies the ablation of the module for obtaining editing regions based on word frequency. In this case, editing regions are solely obtained through the LLM, while the other modules remain unchanged.

Table 4 demonstrates the roles played by various key components in our method. We observe that the implicit masking module and the discriminator module have a significant impact. This is because the Implicit separation module implicitly marks the style words in the masked sequences x' and x'' using '[]' instead of directly replacing the style words with [MASK] or removing them altogether. This approach hints at the content and theme of the input text, aiding the model in preserving more information. The discriminator serves as the sole signal for determining the validity of the masked sequences x' and x'', it is equally crucial for our model. Without the discriminator module, we would be unable to assess the correctness of the edited regions generated by the model.

The influence of sample quantity on model performance tested on the Yelp dataset. From Figure 5, we observed that in the case of 0 samples,

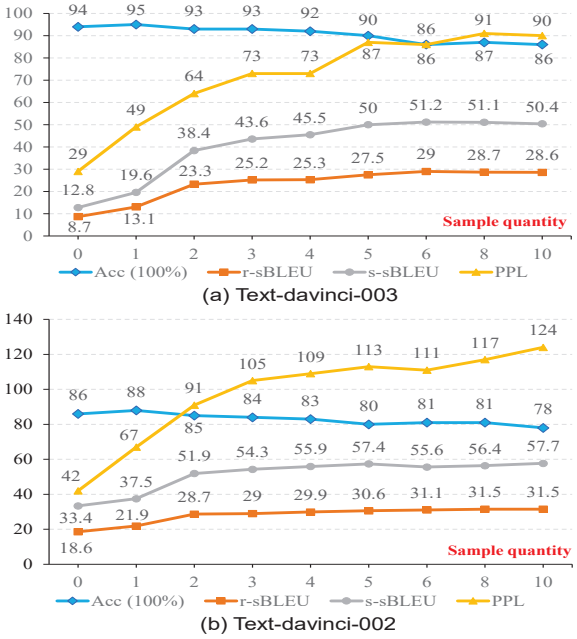

Figure 5: Influence of sample quantity on model performance in prompt templates.

Model	Negative→Positive
Input 1	There is definitely ***not enough*** room in that part of the venue.
LLM Aug	There is ***plenty of*** room in that part of the venue.
P&R	The venue ***has enough space.***
Ours PEGF	***Masking sequence:*** There is definitely ***[not] [enough]*** room in that part of the venue. ***Output:*** There is definitely ***plenty of*** room in that part of the venue.
Input 2	The food was ***low quality*** and thrown together quickly with ***little*** flavor.
LLM Aug	the food was ***high quality*** and ***carefully prepared*** with ***delicious*** flavor.
P&R	The food was ***delicious*** and ***prepared with a lot of care!***
Ours PEGF	***Masking sequence:*** The food was ***[low]*** quality and thrown together quickly with ***[little]*** flavor. ***Output:*** The food was ***high*** quality and thrown together quickly with ***great*** flavor.
Input 3	I am ***only*** giving this item num_num stars because of the ***absurd*** price.
LLM Aug	I am ***awarding*** this item num_num stars ***for its great value for money.***
P&R	I am only giving this item 5 stars because ***it works perfect and the price is fine.***
Ours PEGF	***Masking sequence:*** I am ***[only]*** giving this item num_num stars because of the ***[absurd]*** price. ***Output:*** I am giving this item num_num stars because of the ***reasonable*** price.

Figure 6: Qualitative examples of sentiment transfer. We manually used bold and italics to highlight the stylistically relevant phrases we discovered.

the model exhibited excellent accuracy and fluency but lower content retention. As the number of samples in the prompt template gradually increased, the model's content retention performance significantly improved, but accuracy and fluency declined. The most noticeable difference was between the 0-shot and 2-shot methods.

Case study. In Figure 6, we present examples of the outputs generated by our model and partial outputs from baseline models under the same input conditions. It can be observed that previous approaches, which employ autoregressive generation, suffer from error accumulation, leading to less controlled and unsatisfactory sentence generation.

However, our prompt-based editing method for TST achieves the desired results by identifying

and modifying the text within the editing region. This method selectively modifies a small number of stylistic words without altering other content, thereby preserving more information while accomplishing TST. Moreover, this explicit replacement of stylistic words enables a more observable process for the LLM in performing TST, making it more interpretable and controllable.

5. Conclusion

In this paper, we propose a new prompt-based specific region editing approach for arbitrary text style transfer, which transforms the task of TST into a text filling task. Unlike other autoregressive generation methods, PEGF guides the LLM to edit a small amount of text within specific semantic regions. It selectively modifies a few stylistic words without altering other content, preserving more contextual information, reducing the accumulation of errors in LLM, and enhancing its controllability, stability, and interpretability. Experiments conducted on several publicly competitive datasets for text style transfer task confirm that our proposed approach achieves state-of-the-art performance.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants U22B2036, and 62202381, in part by Shenzhen Science and Technology Program and Guangdong Basic and Applied Basic Research Foundation (2021A1515110717, 2024A1515010087), General Program of Chongqing Natural Science Foundation (No. CSTB2022NSCQ-MSX1284), Sponsored by CAAI-Huawei MindSpore Open Fund, the National Postdoctoral Innovative Talents Support Program for L. Wu. We would like to thank the anonymous reviewers for their constructive comments.

7. limitations

Our paper introduces the transformation of TST tasks into text infilling tasks, achieved by guiding the LLM to edit text within specific regions to accomplish TST. However, based on our studies, we have identified the following main limitations:1) Multi-turn overhead issue: We observed that compared to other LLM approaches (Reif et al., 2022) that regenerate target-style text through autoregressive generation, our model achieves TST by making minimal modifications within the editing region. However, it suffers from the overhead of multiple turns. For instance, our model has multiple outputs, and evaluating multiple outputs requires multiple computations before selecting the output with the highest score as the final output.2) Manual prompt template design: The current approach in the field of TST relies heavily on manually designing prompt templates. However, this method relies on prior knowledge and requires a significant amount of time. Additionally, it is challenging to design an optimal prompt template. In the future, we aim to alleviate this issue by exploring automatic template learning techniques (Liu et al., 2023).

8. Ethics Statement

In the realm of textual style transfer, as with numerous applications in Natural Language Processing, it's crucial to consider the ethical implications associated with our advancements. While our research strives to enhance the capabilities of arbitrary textual style transfer, it is essential to recognize the potential misuse of these methods by individuals with malicious intentions.

For instance, one conceivable misuse could involve transforming a neutral news headline into a sensationalized version or altering a harmless text into an offensive message, all aimed at fulfilling a harmful objective.

These possibilities underscore the importance of responsible innovation and usage. Ethical considerations should guide the development and deployment of these technologies. Researchers, practitioners, and policymakers must collaborate to establish ethical frameworks, ensuring that these tools are employed for positive and constructive purposes. By fostering awareness and encouraging ethical practices, we can promote a safe and beneficial application of textual style transfer techniques, thus contributing to a more responsible and considerate digital environment.

9. References

Shuguang Chen, Leonardo Neves, and Thamar Solorio. 2022. Style transfer as data augmentation: A case study on named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1827–1841.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuan-Jing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of*

the AAAI Conference on Artificial Intelligence, volume 32.

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C Aggarwal, and Aston Zhang. 2022. Text style transfer: A review and experimental evaluation. *ACM SIGKDD Explorations Newsletter*, 24(1):14–45.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Juncen Li, Robin Jia, He He, and Percy Liang. 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)*, pages 1865–1874.

Ao Liu, An Wang, and Naoaki Okazaki. 2022. Semi-supervised formality style transfer with consistency training. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4689–4701.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Ruibo Liu, Chenyan Jia, Jason Wei, Guangxuan Xu, Lili Wang, and Soroush Vosoughi. 2021. Mitigating political bias in language models through reinforced calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14857–14866.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. *arXiv preprint arXiv:1905.10060*.

Guoqing Luo, Yu Tong Han, Lili Mou, and Mauajama Firdaus. 2023. Prompt-based editing for text style transfer. *arXiv preprint arXiv:2301.11997*.

Daniil Moskovskiy, Daryna Dementieva, and Alexander Panchenko. 2022. Exploring cross-lingual textual style transfer with large multilingual language models. *arXiv preprint arXiv:2206.02252*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 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)*, pages 837–848.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *Advances in neural information processing systems*, 30.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. 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)*, pages 3269–3279.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 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*, pages 2195–2222.

Jiarui Wang, Richong Zhang, Junfan Chen, Jaein Kim, and Yongyi Mao. 2022. Text style transferring via adversarial masking and styled filling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7654–7663.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988.

10. Language Resource References

Juncen Li, Robin Jia, He He, and Percy Liang. 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)*, pages 1865–1874.

Rao, Sudha and Tetreault, Joel. 2018. *Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.