

A Study on Text Style Transfer via
Pre-trained Language Models

事前学習に基づく言語モデルによる
テキストスタイル変換に関する研究

山梨大学大学院
医工農学総合教育部
博士課程学位論文

2025年3月

徐勝

Contents

1	Introduction	1
1.1	Text Style Transfer	2
1.1.1	Definitions of Style	3
1.1.2	Research Scope of TST	3
1.2	Approaches on TST	5
1.2.1	Training from Scratch	5
1.2.2	Pre-training and Fine-tuning	7
1.2.3	LLM-based Prompting	8
1.2.4	Innovations of Our Approaches	8
1.3	Applications of TST	9
1.3.1	Research-oriented Applications	9
1.3.2	Human-computer Interaction Scenarios	10
1.4	Thesis Outline	11
2	Related Work	13
2.1	PLMs-based Approaches	13
2.1.1	Fine-tuning	13
2.1.2	Prompting	15
2.2	Datasets and Benchmarks	18
2.2.1	Annotated Datasets	18
2.2.2	Data Augmentation	19
2.3	Metrics for Evaluation	20
2.3.1	Human Evaluation	20
2.3.2	Automatic Evaluation	20

2.3.3	LLMs-based Evaluation	21
3	Incorporating Sentiment Transfer and Paraphrase Generation for PTR	23
3.1	Motivation	23
3.2	Methodology	25
3.2.1	Decomposing PTR	25
3.2.2	Creating Pseudo Data as Prior Knowledge	25
3.2.3	Fusion Strategies	27
3.3	Experiments	27
3.3.1	Experimental Setup	27
3.3.2	Results	28
3.3.3	Case Study	29
3.4	Summary	31
3.4.1	Contribution	31
3.4.2	Limitation	31
3.4.3	Future Work	32
4	Disentangling Style from Content by Contrastive Learning for PTR	33
4.1	Motivation	33
4.2	Method	34
4.2.1	Seq2Seq Text Generation	35
4.2.2	Decoupling Style from Contents	35
4.2.3	Preserving Primary Content	36
4.3	Experiments	37
4.3.1	Experimental Setup	37
4.3.2	Results	38
4.3.3	Ablation Study	38
4.3.4	Case Study	40
4.4	Summary	41
4.4.1	Contribution	41
4.4.2	Limitation	41
4.4.3	Future Work	42

5	Rethinking SST as Reduction and Synthesis	43
5.1	Motivation	43
5.2	Methodology	45
5.3	Plug-and-Play Approach	45
5.3.1	Problem Formulation	46
5.3.2	Reduction and Synthesis	46
5.3.3	Self-Refrine Augmented RS	47
5.4	Experiments	47
5.4.1	Setup	47
5.4.2	Evaluation Metrics	48
5.4.3	Results	50
5.4.4	Case Study	53
5.5	Summary	54
5.5.1	Contribution	54
5.5.2	Limitation	54
5.5.3	Future Work	55
6	Static and Dynamic Prompting for SST	57
6.1	Motivation	57
6.2	Methodology	58
6.2.1	Dynamic Prompt Template	58
6.2.2	Self-Refine Augmented Dynamic Prompting	59
6.3	Experiments	60
6.3.1	Experimental Setup	60
6.3.2	Evaluation Metrics	60
6.3.3	Results	62
6.4	Summary	66
6.4.1	Contribution	66
6.4.2	Limitation	66
6.4.3	Future Work	66
7	Entangle or Disentangle, which is Better for TST	69

7.1	Motivation	69
7.2	Method	70
7.2.1	Constructing Pipelines by Leveraging CoT	70
7.2.2	Assemble Disentanglement and Entanglement	71
7.3	Experiments	72
7.3.1	Experimental Setup	72
7.3.2	Results	73
7.4	Summary	76
7.4.1	Contribution	76
7.4.2	Limitation	76
7.4.3	Future Work	77
8	Conclusion	79
8.1	Dataset Perspective	79
8.2	Model Perspective	80
8.3	Evaluation Perspective	80
	Publications	82
	Appendices	97
A	Formalization of Reduction-and-Synthesis	97
B	Hyperparameter	97
C	Prompt Templates	98
D	Demonstration	106

SUMMARY OF THESIS

This thesis focuses on five PLM-based approaches for the Text Style Transfer (TST) task. In any language, text serves as a medium for recording and transmitting information. From a macro perspective, text can be categorized into style and style-independent content. Content typically refers to the core message we intend to convey. At the same time, style pertains to how the content is expressed, such as whether the tone is positive/negative or formal/informal. Therefore, all TST tasks can be interpreted as transforming an input text “X” that represents content “C” with an original style “ a ”, into an output “X’” that preserves “C” while adopting a target style “ a' ”. TST can include a range of sub-tasks, such as Positive Text Reframing (PTR), Sentiment Style Transfer (SST), and Formality Style Transfer (FST), depending on the type of style pair (a, a') involved.

Building upon a review and analysis of the existing research in TST, this thesis proposes two methods for fine-tuning and three studies based on prompting techniques. Through experimental analysis and case studies, we summarize the advantages and limitations of each approach and promising directions for future research.

Chapter 1 introduces the definition of text and TST, and surveys three main approaches in previous research based on Deep Neural Networks (DNNs): 1) training from scratch, 2) fine-tuning, and 3) prompting. We also provide a brief overview of the applications of TST in various scenarios.

Although the first type of paradigm, training from scratch, theoretically offers a more direct simulation of a conditional generative model constrained by input sentences, its effectiveness is limited by the lack of access to large-scale parallel data samples. Consequently, the model struggles to achieve robust performance across complex and diverse TST sub-tasks. To effectively utilize large-scale unlabeled text and capture the intrinsic distribution patterns of text sequences, such as sentences or documents, pre-trained language models (PLMs) based on Transformer architectures have become a practical solution. Notable small-scale models, such as BERT and GPT, acquire generalizable features of text sequences through training on extensive datasets, thereby achieving strong generality and adaptability. These models can then be fine-tuned on small parallel datasets to facilitate controllable text generation for specific TST sub-tasks. Considering their strong generalization capabilities, PLMs have taken a dominant position in addressing NLP tasks. Therefore, this thesis primarily investigates PLM-based approaches for TST tasks.

To demonstrate the innovation and motivation of our five research works, we first, in Chapter 2, review and summarize the previous PLMs-based approaches to TST tasks before presenting each of our works in detail. By presenting an in-depth analysis of several representative studies, with a detailed examination, we set the foundation for the subsequent discussion from Chapter 3 to Chapter 7.

We first introduce two different fine-tuning methods for applying PLMs to a specific TST subtask, PTR, in Chapters 3 and 4. In Chapter 3, following the decomposition of the PTR task, we introduce a novel fine-tuning method and a data augmentation strategy for auxiliary tasks. Unlike previous methods that directly perform end-to-end fine-tuning of PLMs using parallel datasets to model the PTR task, we first analyze the PTR task and decompose the transformation process into two sub-tasks: paraphrase generation and style transfer. Furthermore, to enhance the model’s performance in these sub-tasks, we constructed two pseudo-datasets using existing paraphrasing and sentiment datasets for further fine-tuning of the PLMs.

This design offers two advantages: 1) Due to extensive attempts, high-quality parallel paraphrasing datasets and annotated sentiment datasets are more readily available than parallel datasets for the PTR task; 2) by using a two-step process, the model can generate more diverse and varied sentences. Through experimentation, we validated the effectiveness of this task-level decomposition approach.

While decomposing the PTR task at the task level provides the aforementioned advantages, the overall performance of the model may be influenced by the quality of the two pseudo-datasets. If the data used for training the auxiliary tasks is of low quality, it may adversely affect the PLM’s performance on the PTR task. To this end, in Chapter 4, we propose a new disentangling training objective based on fine-tuning to mitigate potential errors introduced by pseudo-parallel data augmentation and to rely only on the PPF dataset.

Furthermore, to enable the model to learn the more precise and fine-grained data features implicitly embedded in the PTR sentence pares, this approach uses contrastive learning to constrain the hidden space of the Transformer model, effectively separating style representations from content hidden states, and controlling the generation of sentences with the target style representations. Through experiments, we found that our fine-tuning method improved the style transfer strength of the model on two different PLMs (BART and T5). Compared to the baseline, the model was able to generate more fluent sentences.

While the large language models (LLMs) are inherently a special case of PLM, their superior generalization ability offers significant advantages compared to fine-tuning large models, which are more computationally expensive. In contrast, prompting-based methods can efficiently improve overall model performance. Moreover, since these methods do not rely on parallel datasets, LLMs based on prompting can be applied to a wide range of downstream tasks. Therefore, in Chapters 5 through 7, we explore three different prompting strategies in TST sub-tasks such as SST, and FST.

By leveraging the fine-tuning method, we can guide the model to learn separate representations of style and content, or the overall sentence representation, in the hidden space by applying constraint functions during the training process. However, we are unable to directly control the transfer process through linguistic forms. Prompting

methods based on LLMs provide a feasible approach to directly constrain the model’s generation process via prompts. To explore the performance of this approach, we design a prompting pipeline for the SST task in Chapter 5.

We first analyze the characteristics of the SST task and find that for the same model or prompting pipeline the difficulty of transferring varies along the diversity of input sentences. Especially, we define an intuitive method to assess the transfer difficulty of a specific case. For the more challenging transfer cases, we decompose the language model’s operations into two steps, reduction and synthesis (RS). To implement our idea, we propose a novel Plug-and-Play strategy. The experiment results on two popular SST datasets, Yelp and Amazon, demonstrate that our RS pipeline improves the style transfer strength of the baseline model in more complex transfer cases.

Although the RS pipeline improves the model’s ability to handle complex inputs, it requires more inference time compared to the baseline, which only involves a single prompting step. Specifically, when self-refine is used to further enhance the model’s overall robustness, the time complexity limitations of the RS pipeline are further amplified. In Chapter 6, we refocus on the prompt template and aim to enhance the baseline’s performance on the SST task by improving the quality of the prompt templates. To this end, based on a review of previous prompting strategies, I explored the usage of the aspect-based sentiment analysis (ABSA) model for constructing dynamic prompt templates. This strategy mitigates the limitations of static prompts, which often fail to adapt accurately and effectively to diverse input cases. Similarly, we also investigated a new variant of dynamic prompting, in which the ABSA model is applied to the self-refinement algorithm to dynamically construct refinement prompting templates. In contrast to the use of self-refine in Chapter 5, the feedback is predicted by the ABSA model rather than generated by the LLM based on the examples provided in the prompt.

Considering the impact of the type of backbone model, the model size, and the template style on different prompting pipelines, we comprehensively compare and analyze static prompting and dynamic prompting across different LLMs and prompt templates for the SST task. Through extensive experiments, we found that dynamic prompting improves the model’s style transfer strength compared to static prompting. Furthermore, this finding holds across different models and template styles.

The approaches introduced from Chapter 3 to 6 focus on a specific TST sub-task related to sentiment. To fully harness the generalization capabilities of LLMs, we further investigated the performance of LLM-based prompting methods across four different TST sub-tasks in Chapter 7. Based on two fundamental perspectives, disentanglement, and entanglement, we utilize a chain-of-thought (CoT) approach to directly decouple or couple the content and style components of the input sentence at the linguistic level. We then design two prompting methods to implement the disentangle and entangle pipelines.

Disentanglement and entanglement are two general strategies used to address TST problems, and CoT-based prompting for TST has also been explored. However, comprehensive comparison and analysis of disentanglement and entanglement strategies using LLM-based CoT pipelines remains insufficient. Moreover, unlike other related works that focus solely on either the disentanglement or entanglement strategy, in Chapter 7, we employ an LLM-based scorer to ensemble the results generated by both strategies, thereby achieving the best trade-off performance for arbitrary style transfer and content preservation. Through experiments across seven datasets for TST subtasks, we analyze the performance of these strategies both individually and in combination and demonstrate their effectiveness.

Lastly, in Chapter 8, we present a comprehensive summary of the five research studies discussed above. Through analysis from three distinct perspectives (dataset, model, and evaluation), we systematically assess their strengths, weaknesses, contributions, and limitations, while identifying promising directions for future research.

Acknowledgements

First, I would like to express my sincere gratitude to Professor Fumiyo Fukumoto. Since 2019 as a master's student, I have been studying at the University of Yamanashi, she supervised and cared for me quite carefully. She taught me to do experiments and submit my first SCI paper in the NLP research area. During my Ph.D. course, she consistently supports me in overcoming any challenges with my studies, research, and daily life. For my thesis, she gave me a lot of comments and made careful revisions. Under her guidance, I became more confident, finished my Ph.D course, and gained a desired job offer. Let me convey my sincerest gratitude to her.

I also want to sincerely appreciate Professor Yoshimi Suzuki. He provided a quite professional and convenient environment to conduct massive experiments. Moreover, when I met difficult problems in study and daily life, he always patiently helped and encouraged me to gradually overcome these challenges. He taught me a lot of useful skills and knowledge.

Furthermore, I am grateful to Associate Professor Li for the significant support and guidance he provided throughout my doctoral research. I would like to thank Professor Harimoto, Professor Go, and Professor Nishizaki, who have given me valuable feedback on my thesis, which greatly contributed to the successful completion of my work.

I am also thankful to have gotten a lot of help from Pantitan, a senior Ph.D. student in my lab. He gave me many useful advice and resolutions when I met troubles. He is also good at outdoor activities and invited me to hike in Yamanashi during the holiday. I am happy and enjoyed studying and playing with him.

I would like to express sincere thanks to my family. They always approve of my decisions and career plans. Without their understanding, support, and care, it would be impossible to overcome difficulties and take my Ph.D. degree.

Finally, I am deeply grateful to the University of Yamanashi for providing an excellent research and learning environment. Furthermore, I am also honored to be a candidate for the project, “山梨大学融合研究実践ドクターフェローシップ (JST SPRING, Grant Number JPMJSP2133)”, and JASSO scholarship.

Chapter 1

Introduction

Since the early foundational works in the 1950s, such as Turing Test ([Turing, 1950](#)) and Universal Grammar ([Chomsky, 1957](#)), research and technology in Natural Language Processing (NLP) have made significant advancements. With the introduction of ChatGPT issued by [OpenAI](#) in November 2022, the Large Language Models (LLMs) have demonstrated remarkable capabilities of NLP in applications from domain-specific question-answering systems to general-purpose chat robots.

Owing to the enhancement of computational power, advancements in models and algorithms, and the accumulation of vast corpora, NLP has expanded from classical applications such as syntactic parsing and machine translation to encompass more diverse, fine-grained, and complex branches. It is worth noting that research on language features associated with human-like characteristics, such as sentiment analysis and stylistic analysis, has gained increasing popularity recently.

The incorporation of prose style into natural language generation was first considered as early as 1985 in the study of [McDonald and Pustejovsky \(1985\)](#). Unfortunately, due to the limitations of model capability and insufficient parallel corpora, controllable language generation related to stylistic features did not regain significant attention until the emergence of neural language models. For example, [Hu et al. \(2017\)](#) proposed a generative model that reliably generates sentences from hidden states by controlling the stylistic attributes. In section 2, we will introduce more details about those deep learning-based methods for controllable text generation, and TST.

Following style transfer has been extensively studied in computer vision ([Gatys et al., 2016](#); [Li et al., 2017](#); [Huang and Belongie, 2017](#)), Text Style Transfer (TST), as the similar research in NLP area, was formally explored by [Shen et al. \(2017\)](#). Subsequently, particularly with the widespread exploitation of Pre-trained Language Models (PLMs) and LLMs, TST has garnered increasing attention from researchers and has become a subdomain of NLP, as shown in Figure 1.1 which is introduced by [Schopf et al. \(2023\)](#).

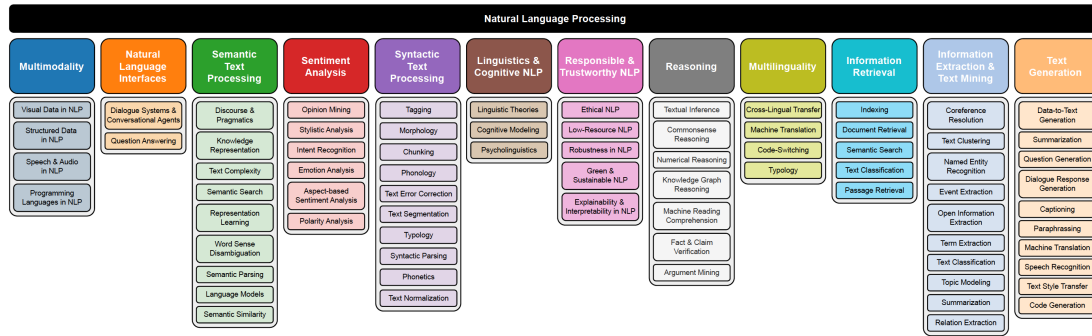


Figure 1.1: An taxonomy of fields of study in NLP induced by [Schopf et al. \(2023\)](#)

To facilitate further research, this thesis introduces the theories and developments of TST, summarizes related work in this area, and illustrates five studies that, without loss of generality, focus on transferring sentence-level text.

In the following section, we will first explain the stylistic features of text and the research scope of TST, then give an overview of different methodological approaches for this task, and introduce several valuable application scenarios by utilizing TST in section 1.2 and section 1.3 respectively. In the final section of this chapter, we will provide a brief overview of the organization of this thesis.

1.1 Text Style Transfer

The text style is the core notion in research related to TST. Therefore, let's first define the concept of style of text. Two paragraphs briefly outline the definition of NLP, as follows:

p1-1 Natural Language Processing (NLP) is how computers learn to understand and use human language. It helps them read, talk, and answer questions. For example, NLP makes tools like chatbots and translation apps work. These tools can communicate with us like people do.

p1-2 Natural Language Processing (NLP) is a field of study that enables computers to understand and utilize human language effectively. It supports various tasks, including text comprehension, speech generation, and question answering. For instance, NLP underpins applications such as chatbot and translation tools, enabling them to engage with users and provide intelligent, context-aware assistance.

As outlined by [McDonald and Pustejovsky \(1985\)](#), style is an intuitive notion involving how something is said. Both examples elucidate the concept of NLP, however,

the description in the first paragraph is more accessible, whereas the second paragraph offers a more scholarly definition.

1.1.1 Definitions of Style

Although the two examples above provide an intuitive demonstration of text style, a more thorough investigation necessitates a precise definition of what text style is, and how to measure it (Heylighen et al., 1999). Jin et al. (2022) concluded the definitions of text style from two perspectives: linguistic features and data characteristics.

- (1) **Linguistic Perspective.** In the scope of linguistics, text style can be understood as the inherent manner and preference through which individuals use language, including rhetorical devices, word selection, sentence construction, and essay composition. Thus, text style is individualized varies across different individuals, and is diverse for even the same speaker as explored by Allan and Adam (1997). On the other hand, it can also be regarded as a set of standardized linguistic rules that strictly define and categorize each style associated with a specific language. For instance, Hovy (1987) proposed three stylistic notions (formality, haste, and force) to guide text generation by incorporating pragmatic information.
- (2) **Data Perspective.** We can also define the text style based on corpora characteristics. By assuming that the dataset inherently encapsulates text styles, text style is determined by predefined label categories within the dataset. In this context, linguistic concepts are primarily employed for human evaluation during experiments and for manual data annotation. As an illustration, for the reviews in Yelp Open Dataset¹, Xiang et al. (2015) defined the sentiment label of each review by considering stars 1 and 2 negative, and 3 and 4 positive. Similarly, Shen et al. (2017) classified reviews with ratings above three as positive text, while those with lower ratings were categorized as negative text.

The potential variations in text style are extraordinarily vast, particularly when studying across different genres, registers, or domains. Therefore, to define the quantification of the dimension of style, some attempts focused on just one aspect of style, such as formality (Heylighen et al., 1999; Brooke et al., 2010; Mosquera and Moreda, 2011), and sentiment (Shen et al., 2017; Ziems et al., 2022).

1.1.2 Research Scope of TST

Early works on text generation related to linguistic style were predominantly to control the generation process by incorporating predefined text style (McDonald and Puste-

¹<https://www.yelp.com/dataset>

jovsky, 1985; Hovy, 1987; Xu et al., 2012). With the advent and widespread application of deep learning techniques in NLP, TST has evolved into a distinct research domain (Mueller et al., 2017; Shen et al., 2017; Jhamtani et al., 2017).

In essence, given an input text and a specified target style, TST involves generating text that aligns with the target style while preserving the style-independent content of the input text. Note that unless explicitly specified otherwise, the “style-independent content” will be referred to simply as “content,” a convention adhered to throughout the remainder of this thesis. Therefore, there are two critical challenges as follows:

- Developing effective representations for both style and content of the input text.
- Combining the content with the target style to generate target text.

In section 1.2 we briefly introduce previous attempts that tackled these two challenges through three distinct methodological paradigms. The more details of the related works are explained in chapter 2.

The primary research on TST aims to address the two critical challenges by exploring the algorithms, models, or frameworks. To provide a more nuanced understanding of the scope of these works, We present the taxonomy based on four criteria: style category, language granularity, language type, and model generality.

- (1) **Style.** Based on the types of styles processed, TST can be further categorized into a series of subtasks. These include formality style transfer (FST), which focuses on the formality of text; sentiment style transfer (SST), which deals with sentiment-related transformations; positive text reframing (PTR), which is similar to SST but prevents the reversal of the sentence’s meaning to the opposite when transferring the sentiment; and authorship style transfer (AST), etc.
- (2) **Granularity.** From the perspective of textual granularity, TST can also be classified into text transfer at the sentence level and those targeting the paragraph/document level.
- (3) **Language.** Furthermore, depending on the corpus or dataset, TST can be applied to process various languages, such as English, Chinese, and Japanese.
- (4) **Generality.** Much of the previous research has focused on individual basic styles, language granularities, or language types. Simultaneously, some pioneering studies have also explored arbitrary TST, multi-style transfer, and multilingual TST. The generality of the approaches or methods employed can be leveraged to classify various types of TST.

In addition to the numerous studies outlined above, which focus on algorithms and models for the ontology of TST, there are also other noteworthy contributions, including the development of datasets (Jhamtani et al., 2017; Rao and Tetreault, 2018;

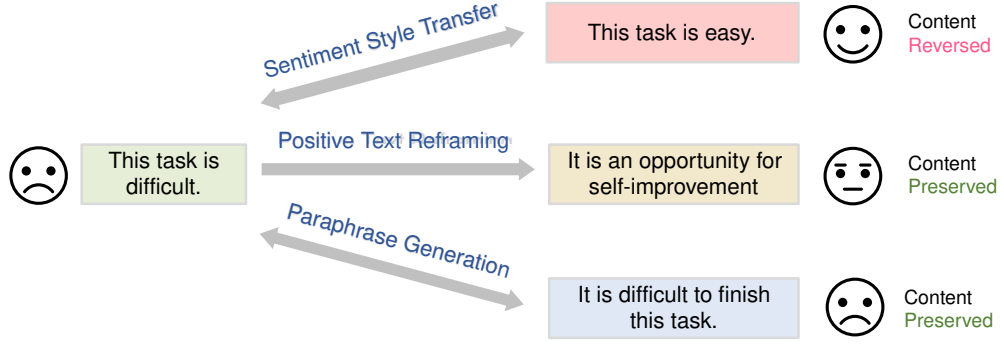


Figure 1.2: A comparison of three similar tasks, SST, PTR, and PG

Pryzant et al., 2020; Wang et al., 2020a; Zhu et al., 2023; Mukherjee et al., 2024b; Sun et al., 2024), the formulation of evaluation methodologies (Fu et al., 2018; Tikhonov et al., 2019; Jafaritazehjani et al., 2020; Briakou et al., 2021; Lai et al., 2023; Os-theimer et al., 2024), and the exploration of applications (Zhu et al., 2021; Wei et al., 2023; Fukasawa et al., 2023).

In this thesis, without loss of generality, we focus on the research about the sentiment attributes of text, that is, PTR and SST subtasks. To illustrate the differences between PTR and SST, we employ the Paraphrase Generation (PG) task, which solely generates an alternative of the input while maintaining its original meaning, as a benchmark for reference. As shown by the example in Figure 1.2, for the same input “*This task is difficult*”, the target sentence transferred by SST with positive style is “*This task is easy*”, while the target of PTR is “*It is an opportunity for self-improvement*”.

PTR places greater emphasis on content preservation (from “*difficult*” to “*opportunity for self-improvement*”). In contrast, SST involves reversing the meaning of the source text as part of the sentiment transfer process (from “*difficult*” to “*easy*”). On the other hand, both PG and SST support bidirectional transfer, whereas PTR modifies the input in only one direction.

1.2 Approaches on TST

As mentioned in section 1.1, the domain and scope of TST have been formally established in the research primarily leveraging deep learning methods since 2017. In this section, we introduce three distinct methodological paradigms and outline previous approaches attempted based on each paradigm for TST.

1.2.1 Training from Scratch

With the initial introduction of neural network (NN)-based language model by Bengio et al. (2000), the neural language models and deep learning methods, have progres-

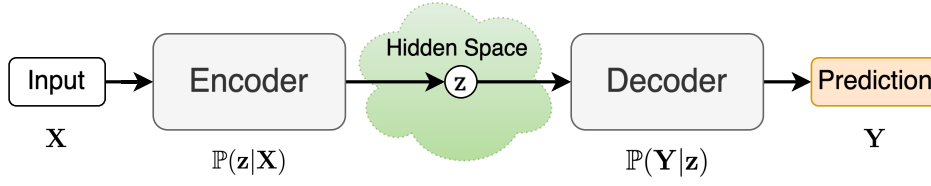


Figure 1.3: Encoder-Decoder Architecture for Training Language Model from Scratch

sively dominated research in various NLP tasks. For sequence-to-sequence (Seq2seq) tasks, including TST, researchers primarily adopt the encoder-decoder architecture for task modeling. In this structure, the encoder and decoder can be implemented by using various models, such as Recurrent Neural Network (RNN)(Mikolov et al., 2010), Convolutional Neural Network (CNN) (Kalchbrenner et al., 2014), and Transformer (Vaswani et al., 2017).

As depicted in Figure 1.3, input text (X) is transformed by encoder, $\mathbb{P}(z|X)$, into a latent space representation (z), which contains both content and style features. The decoder then learns the mapping from this latent representation to the output, $\mathbb{P}(Y|z)$. For labeled text data, supervised learning methods can be used to train task-specific models from scratch for each TST subtask such as FST (Rao and Tetreault, 2018; Xu et al., 2019; Zhang et al., 2020b) and AST (Jhamtani et al., 2017). However, a critical limitation of this approach is its heavy dependency on expensive labeled parallel data.

To overcome this limitation, researchers have proposed improved deep neural network models, such as Variational AutoEncoder (VAE) (Kingma and Welling, 2022), and Generative Adversarial Network (GAN) (Goodfellow et al., 2014), which have achieved significant progress in a wide range of generative tasks. These models, capable of being trained directly on unlabeled text data, have also enabled the application of a wide range of unsupervised learning methods to TST tasks (Shen et al., 2017; Hu et al., 2017; Zhao et al., 2018; Liu et al., 2020; Li et al., 2020).

Those unsupervised learning-based methods primarily consist of two groups. The attempts from the first group assume the input sentence can be encoded into content and style representations in hidden space. By disentangling style representation from content and designing carefully objective functions to constrain the latent state (z), the encoder-decoder architecture can be controlled to transfer diverse text styles without parallel datasets (Mueller et al., 2017; Hu et al., 2017; Liao et al., 2018; Lee et al., 2021). Rather than disentangling content and style representations in the latent space, an alternative approach from the second group involves directly injecting controllable style representations into the entangled hidden state of the source sentence, providing another foundation for the design of unsupervised models (Shen et al., 2017; Fu et al., 2018; Dai et al., 2019; Yi et al., 2020). Furthermore, Han et al. (2023) introduced a transfer pattern mining approach, enabling the model to learn underlying transfer patterns and derive more accurate sentence representations for unsupervised training.

Going beyond the two groups mentioned above, researchers have also introduced

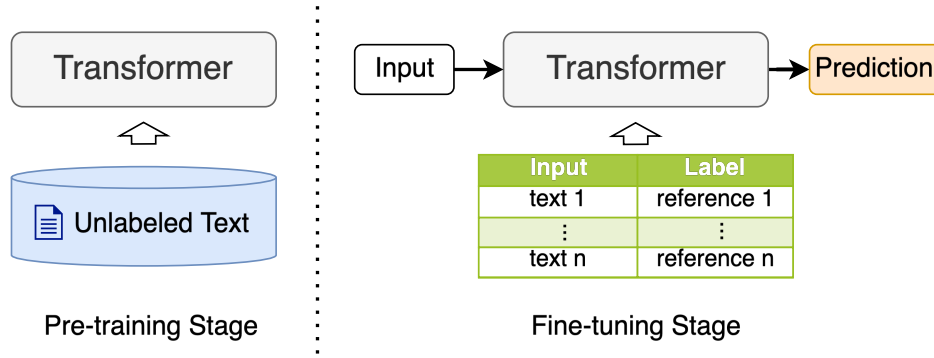


Figure 1.4: Pre-training and Fine-tuning Architecture for

several innovative strategies. For example, [Li et al. \(2018\)](#) framed TST as a process involving delete, retrieval, and generation, employing distinct modeling techniques for each stage. [Huang et al. \(2021\)](#) proposed a general non-autoregressive learning framework tailored for TST. [Lyu et al. \(2023\)](#) leveraged diffusion learning ([Ho et al., 2020](#)) to reconstruct target sentences from embeddings sampled from a Gaussian distribution.

1.2.2 Pre-training and Fine-tuning

Based on the self-attention mechanism ([Vaswani et al., 2017](#)), in sequence modeling and handling long-term dependencies, Transformer models gain outstanding performance. Subsequently, their derivatives (e.g., GPT ([Radford et al., 2018](#)), BERT ([Devlin et al., 2019](#))) trained on large-scale unlabeled data have emerged as dominant architectures in NLP research.

As depicted in Figure 1.4, the downstream tasks can be effectively addressed by fine-tuning the general transformer-based PLMs using relatively small amounts of parallel data. Due to their exceptional performance and generalization capabilities, PLMs can be directly fine-tuned on available parallel TST datasets, and surpass task-specific models that are trained from scratch for individual generation and classification tasks. ([Rao and Tetreault, 2018](#); [Carlson et al., 2018](#); [Lyu et al., 2021](#); [Logacheva et al., 2022](#); [Ziems et al., 2022](#)). To further enhance both content preservation and style transfer performance, the training process of fine-tuning can be combined with advanced machine learning techniques, such as reinforcement learning ([Lai et al., 2021b](#)), and contrastive learning ([Xu et al., 2024](#)).

For non-parallel datasets, parallel pseudo data, which are commonly constructed through data augmentation methods, are subsequently utilized for fine-tuning PLMs ([Lai et al., 2021a](#); [Liu and Chen, 2022](#)). Furthermore, researchers have extensively investigated unsupervised methods based on PLMs ([Li et al., 2022](#); [Wang et al., 2022](#)). For example, [Li et al. \(2022\)](#) employed meta-learning and adversarial training strategies, updating the state-of-the-art (SOTA) performance. [Nouri \(2022\)](#) utilized optimal transport to incorporate syntactic and semantic information simultaneously into the

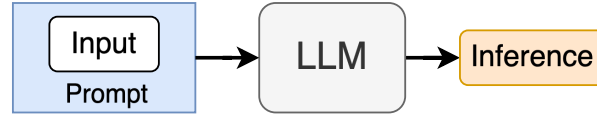


Figure 1.5: Pre-training and Fine-tuning Architecture for

fine-tuning process.

According to the scaling law (Kaplan et al., 2020), the larger models, with a greater number of parameters, theoretically have stronger representation and generalization capacities. As a result, with the expansion of datasets and the increase in model parameters during the pre-training stage, PLMs have achieved groundbreaking performance across numerous downstream tasks. In this case, such PLMs are commonly referred to as large language models (LLMs).

1.2.3 LLM-based Prompting

Based on the scaling laws hypothesis, the GPT-3 (Brown et al., 2020a), which is pre-trained by just scaling up the GPT-2 model (Radford et al., 2019) by a factor of 100, demonstrates exceptional performance across downstream tasks, significantly outperforming smaller models even in the absence of fine-tuning. Subsequently, numerous large language models (LLMs) designed for general NLP tasks have been released, such as LLaMA (Touvron et al., 2023), and Mixtral (Jiang et al., 2024).

As shown in Figure 1.5, the LLMs can be applied to various downstream tasks by formulating the input with a well-designed prompt. Under few-shot or zero-shot settings, massive prompting methods have also been explored for TST (Reif et al., 2022; Suzgun et al., 2022; Yang et al., 2023b; Luo et al., 2023; Liu et al., 2024). As the result, the critical point lies in designing appropriate prompt templates and pipelines that can consistently achieve stable results across various TST subtasks. For the detailed designs and algorithms, we will explain the implementation of several LLM-based prompting-related works in Chapter 2.

1.2.4 Innovations of Our Approaches

Early research on training models from scratch extensively explored methods based on entanglement and disentanglement strategies. These approaches aim to impose constraints on the encoder’s output by explicitly modeling content and style separately. However, such strategies cannot be directly applied to PLMs, as general PLMs are not pre-trained with task-specific objectives for TST. This creates a gap between the carefully designed constraints on hidden representations and the PLM’s original training objectives during fine-tuning.

To address this challenge, in Chapter 3, we introduce a task-level entanglement strategy, where we fine-tune PLMs using pseudo datasets tailored for different tasks. Then, in Chapter 4, we propose an effective disentanglement-based fine-tuning method that relies solely on parallel datasets, leveraging contrastive learning and a carefully designed batch scheduling strategy.

In the era of LLMs, research on TST has primarily focused on designing prompt templates and pipelines or distilling PLMs using LLMs’ powerful generative capabilities. However, these methods have not systematically compared and analyzed the effectiveness of the two fundamental strategies. To bridge this gap, in Chapters 5 and 6, we explore disentanglement-based approaches for SST tasks. In Chapter 7, we systematically compare the performance of entanglement- and disentanglement-based pipelines in terms of style transfer and content preservation. Furthermore, by refining the reranking method proposed by [Suzgun et al. \(2022\)](#), we ensemble the outputs of both pipelines to achieve superior results across four TST subtasks.

1.3 Applications of TST

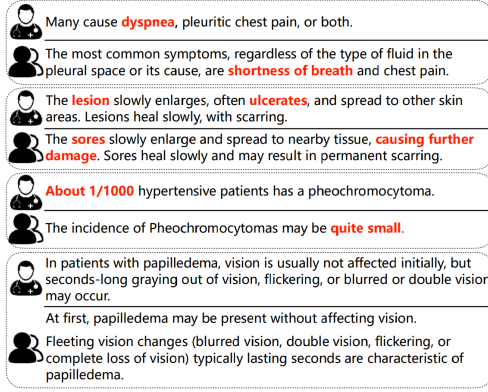
Harnessing the inherent characteristics of language, such as its conciseness, diversity, and rich capacity for abstraction, TST finds extensive applications across a wide range of domains. These applications can be broadly divided into two categories: those focused on advancing NLP and related research areas, and those centered on specific scenarios in human-computer interaction.

1.3.1 Research-oriented Applications

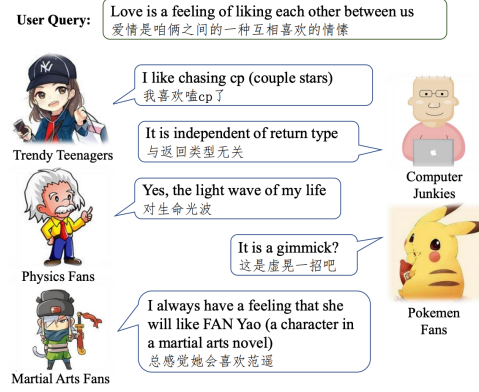
In contrast to foundational NLP tasks such as part-of-speech tagging, tokenization, and syntactic parsing, TST can be regarded as a task at the level of pragmatics which has a wider range of potential applications.

For other NLP tasks, such as question answering and machine translation, TST can be utilized to optimize the systems or train the models. For instance, [\(Wei et al., 2023\)](#) applied TST to the back-translation (BT) task by transforming translation-like inputs into a natural style that reflects text generated by native speakers. This work enhanced the BT model’s capacity to process natural input.

In addition to NLP applications, TST can also be extended to other research topics, such as speech synthesis and computational vision. For example, [\(Zhu et al., 2021\)](#) used TST to augment instruction data and integrated multimodal learning techniques to improve the completion rate of the model for outdoor vision-and-language navigation.



(a) An example of expert style transfer proposed by Cao et al. (2020).



(b) Responses with different characters in personalized dialog Fu et al. (2021).

Figure 1.6: Applications of TST in text simplification and dialog system

1.3.2 Human-computer Interaction Scenarios

In industries such as design, robotics, education, and literature, the application of TST can be further diversified and specialized through human-computer interaction systems based on speech or text. We introduce three examples of applications:

- (1) **Text Simplification.** To enhance comprehension, TST can be employed for text simplification, rendering complex and specialized concepts more accessible to language learners, children, laymen, or individuals with cognitive challenges. As the example shown in Figure 1.6a, Cao et al. (2020) established a benchmark for expertise style transfer to alleviate cognitive biases between experts and laymen during consultations where professional medical terminologies are mentioned.
- (2) **Dialog Systems.** For chatbot applications, various personalized agents can be designed, each with distinct language styles, such as being empathetic, formal, or humorous, thereby offering different user experiences. For instance, Figure 1.6b illustrates the chitchat dialogue systems developed by (Fu et al., 2021). They trained a model to learn the use of jargon, which is subsequently incorporated into the rewriting process. This model ultimately transforms generic conversations into stylized candidates with diverse characteristics.
- (3) **Content Compositing.** TST can also be applied in writing assistant programs (Yeh et al., 2024; Lee et al., 2024). For instance, converting emails into formal business language can assist staff in writing reports. Likewise, the writing assistant can also provide authors with alternative stylistic expressions of the same content in the style of story (Zhu et al., 2023), poetry (Yang et al., 2018a) or Shakespearean literature (Jhamtani et al., 2017).
- (4) **Input Method.** Another meaningful application of TST is its integration into input methods. As designed by Go et al. (2022), when users input text with neg-

ative emotions on mobile devices, PTR can be utilized in real-time to modify the original text into a subjective well-being candidate without altering its original intent. This candidate text can then be recommended to users.

For a more comprehensive exploration of TST applications, we recommend referring to the review by (Mukherjee et al., 2024a).

1.4 Thesis Outline

To facilitate further research, this thesis first surveys the theories and developments of TST summarizes related work in this area, and illustrates five studies, four of which primarily focus on transferring sentence-level text with sentiment style, including PTR and SST, as well as one work which addresses the arbitrary TST subtasks.

Chapter 2 provides an overview of classic approaches to TST based on PLMs and LLMs, offering a detailed analysis and comparison of their respective strengths and limitations.

Chapters 3 through 7 present five original studies conducted as part of this research. Chapters 3 and 7 focus on two distinct methodologies for decomposing content features from the input and integrating target style into style-independent content representations for the PTR task. Chapters 5 and 6 investigate the capacity of LLMs on SST, proposing two prompting strategies to enhance the overall performance of models. In Chapter 7, we expanded the application of LLM-based prompting to encompass other TST subtasks such as FST, and AST.

In Chapter 3, we propose a novel model that integrates paraphrase generation with style transfer, demonstrating its efficacy on the PTR dataset. This approach does not explicitly separate content and style representations. Instead, it focuses on learning to integrate the target style into the content by designing different model architectures.

In Chapter 4, we introduce a fine-tuning method leveraging contrastive learning to capture subtle emotional differences between inputs and outputs, while disentangling content representations from the hidden space. Compared to the method in Chapter 3, this approach places greater emphasis on the separation of content and style representations and the model’s ability to distinguish between different styles through explicit fine-tuning strategies and optimization objectives.

In Chapter 5, To leverage the superior capability of LLMs, we reconsider the formulation of language models in TST, then propose a plug-and-play pipeline to boost the baseline prompt’s performance on two SST datasets, particularly when addressing ambiguous inputs.

In Chapter 6, with the rapid proliferation and iterative advancements of LLMs, we analyze and compare the performance of three widely used open-source LLMs on SST

tasks. Simultaneously, we evaluate the impact of various prompting paradigms on model outcomes.

In Chapter 7, To construct a unified prompting pipeline for arbitrary TST tasks, we reexamined the fundamental characteristics of LLM-based TST using two core strategies: disentanglement and entanglement. Building on this analysis, we developed two distinct pipelines tailored to these strategies and evaluated their performance across various subtasks and LLMs.

Finally, Chapter 8 summarizes the findings of these five studies, discusses the limitations of the proposed models and methods, and highlights several promising directions for further exploration.

Chapter 2

Related Work

This chapter discusses related work on TST, focusing on three primary aspects: 1) PLM-based approaches including fine-tuning and prompting; 2) datasets and benchmarks; and 3) metrics for evaluation.

2.1 PLMs-based Approaches

As briefly explained in section 1.2, for the available parallel datasets fine-tuning on popular PLMs such as GTP-2 (Radford et al., 2019), BART (Lewis et al., 2020), and T5 (Raffel et al., 2020), is the usually utilized to established baselines. To address the scarcity of parallel data for the specific subtasks, the fine-tuning process is often combined with other unsupervised learning strategies. As the size of PLMs increases, for LLMs, substantial computational resources and high-quality annotated data are required for fine-tuning the entire model. In contrast, prompting methods have garnered increased attention from researchers. In the remainder of this section, we will dive into PLM-based related works, focusing primarily on fine-tuning and prompting.

2.1.1 Fine-tuning

Wang et al. (2019) explore three fine-tuning methods that incorporate the preprocessed sentence based on rules into training procedure and achieve a new state-of-the-art on GYAFC, a parallel dataset created by Rao and Tetreault (2018) for FST task. The explanations of these three methods in Figure 2.1 are as follows.

- (1) **Concatenation.** For each input sentence \mathbf{x} , concatenated to a preprocessed counterpart \mathbf{x}' by a special token “[EOS]”, then the new sequence $(\mathbf{x}, [\text{EOS}], \mathbf{x}')$ is leveraged as the input to fine-tune the PLM.

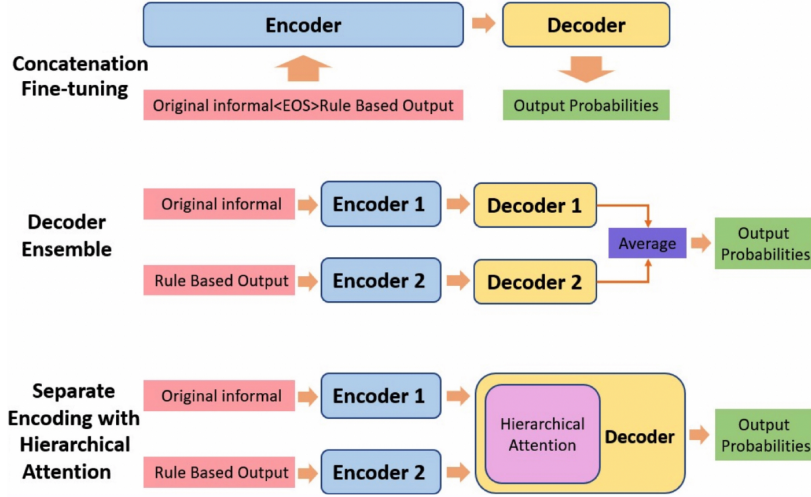


Figure 2.1: Three different methods incorporating PLM with rule-based systems proposed by Wang et al. (2019)

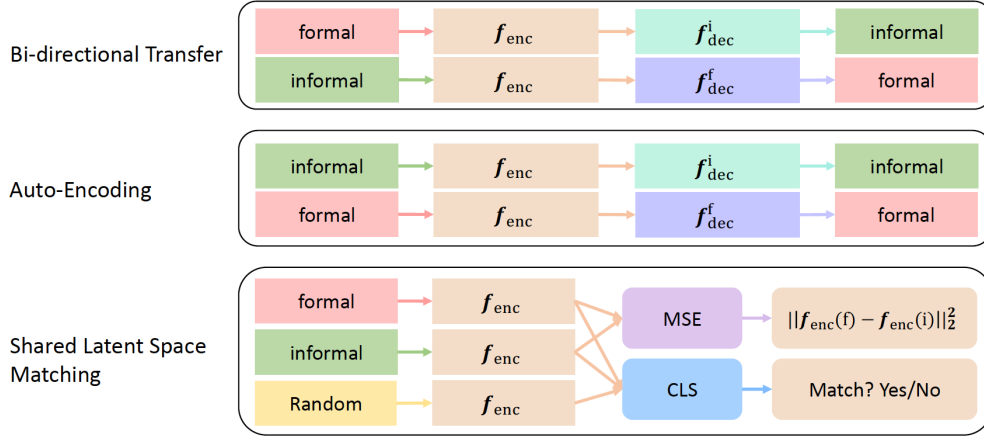


Figure 2.2: The overview of S2S-SLS model proposed by Wang et al. (2020b)

- (2) **Decoder Ensemble.** Two individual PLMs process the original sentence and preprocod counterpart respectively. Then during each generation step, the probability distribution over the vocabulary is averaged by the predictions of the PLMs.
- (3) **Hierachical Attention.** In this variant, the two encoders map the original sentence and preprocod counterpart respectively into hidden states at first. Then, the aggregation is conducted by designing a cross attention for the decoder to combine both states which represent the information of the original sentence and preprocessed counterpart. By experiments, each of these models can improve the performance of PLM on FST compared with previous works.

Wang et al. (2020b) proposed one more novel model which learns multiple auxiliary objectives together with fine-tuning in sharing the hidden states outputted by the encoder. As shown in Figure 2.2, the original fine-tuning objective for the PLM is the loss of the bi-direction transfer pipeline.

To enrich the parallel pairs for training, each pair of formal and informal sentences can also be utilized to reconstruct themselves by the auto-encoding pipeline. Moreover, the two auxiliary tasks are also designed to train the model to achieve two objectives: (1) to learn the semantic distance between the two sentences within each pair, and (2) to identify the true pair from two additional randomly constructed pairs. Throughout training all those six objectives simultaneously by fine-tuning the GPT-2 (Radford et al., 2019), experiments demonstrated the effect of each auxiliary task and the improvement brought by the shared latent space.

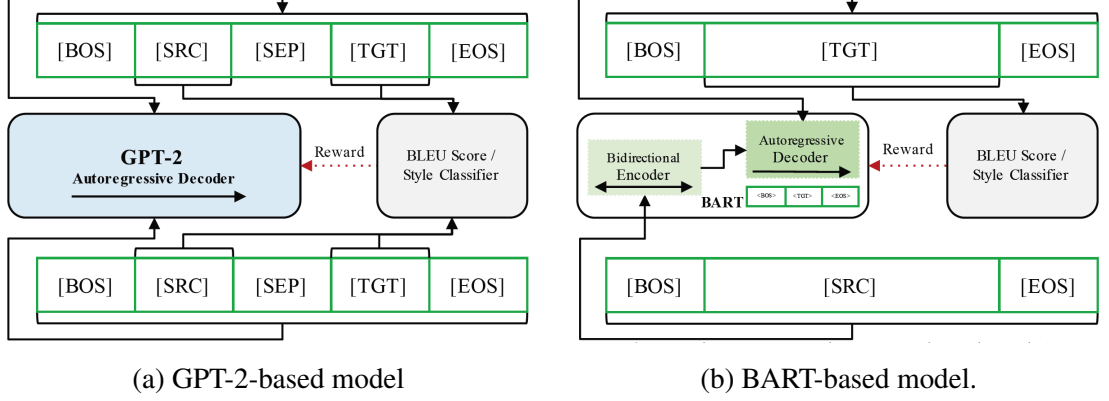


Figure 2.3: Two architectures based on GPT-2 and BART respectively to optimize the reward during fine-tuning (Lai et al., 2021b)

Based on reinforcement learning, Lai et al. (2021b) introduces two rewarding targets to fine-tune the backbone PLMs (GPT-2 and BART) to boost the performance on the parallel dataset GYAFC. The architecture is shown in Figure 2.3, where the reward for preserving content is calculated by the BLEU score between generation (the “[TGT]” segment in the plots of Figure 2.3) and ground truth. Similarly, the transfer strength of PLM is optimized using a style classification reward, corresponding to the probability of the generated output being classified as the target style by a pre-trained binary style classifier. By employing the policy gradient algorithm (Williams, 1992), the two reward scores can be utilized to fine-tune the parameters of PLMs efficiently.

Several other PLM-based studies have also introduced innovative extensions to fine-tuning approaches. For instance, Li et al. (2022) incorporated fine-tuning as a step in meta-learning, addressing the suboptimal performance of PLMs on the datasets with limited labeled data. Similarly, Nouri (2022) employed GPT-2, and fine-tuned the model to learn the interaction between syntactic and semantic similarities in the pairs of source and transformed texts.

2.1.2 Prompting

Prompt learning which is one of the major paradigms of NLP demonstrated outstanding performance for a variety of downstream tasks including text classification (Schick

et al., 2020; Lester et al., 2021), information extraction, text generation (Radford et al., 2019; Schick and Schütze, 2021), and question answering (Jiang et al., 2020). Furthermore, the advancements in large language models (LLMs) over the past few years have attracted significant interest from researchers, especially the progress of the GPT series of models (Radford et al., 2018, 2019).

Benefiting from the significant capacity of LLMs for the TST task, several attempts have explored the discrete prompt-based methods, considering interpretability and controllability.

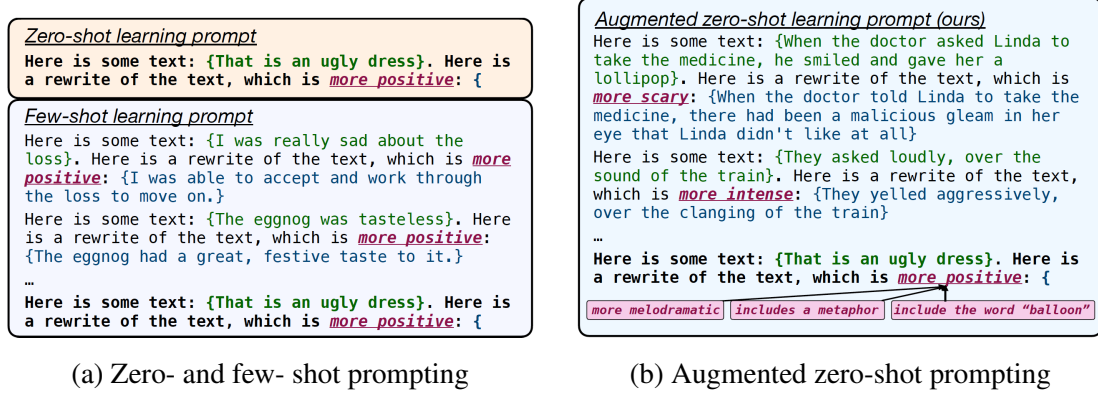


Figure 2.4: Prompting templates for different recipes (Reif et al., 2022)

Reif et al. (2022) proposed a recipe, augmented zero-shot learning, for controllable style transfer using LLMs without updating any parameters. Instead of providing examples from the same transfer task, the augmented prompt, shown in Figure 2.4b, includes a diverse set of transfer examples across different styles. Experiments by leveraging GPT-3 (Brown et al., 2020b) as the backbone for inference demonstrated the augmented prompting recipe outstanding the other templates and the SOTA fine-tuning-based methods.

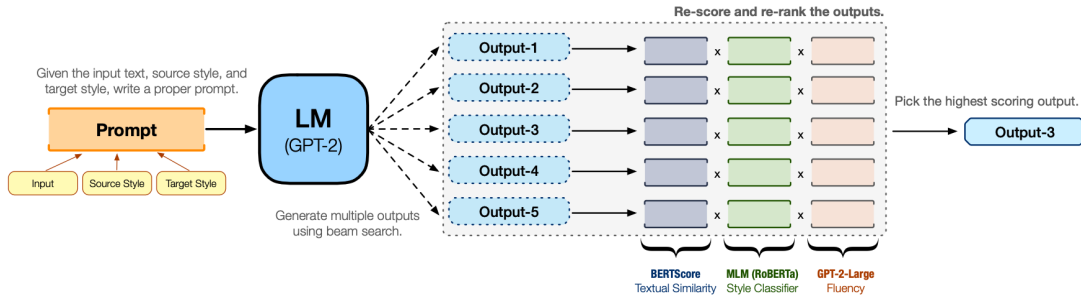


Figure 2.5: Prompt-reranking method for TST (Suzgun et al., 2022)

Suzgun et al. (2022) decomposed the formalized TST task into three main components, i.e., *textual similarity*, *transfer strength*, and *fluency*. They designed a basic prompting template for input sentences and utilized the GPT-2 (Radford et al., 2019) to generate multiple candidates which are re-ranked according to scores of the three components. Finally, the best one will be chosen as the final output. The entire pipeline,

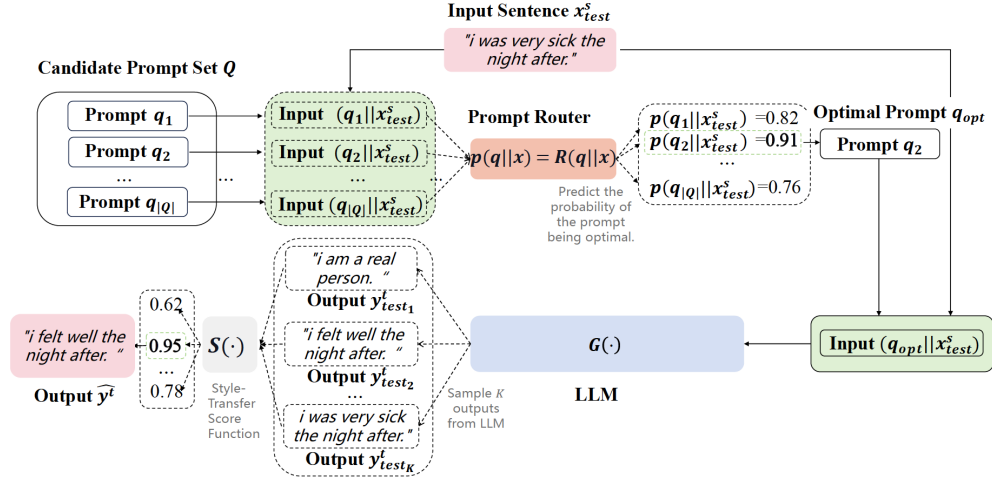


Figure 2.6: Adaptive prompt routing framework for TST (Liu et al., 2024)

prompt-reranking method, is illustrated in Figure 2.5.

In contrast, to overcome the limitation of a single template for all input cases. Liu et al. (2024) constructed a set of prompt candidates ($\{q_1, q_2, \dots, q_{|Q|}\}$, in Figure 2.6), and trained a scoring model (prompt router) that predicts the best template among the candidates which is more likely to produce an appropriate inference for each input.

Other novel prompting-based attempts are not directly leveraged to infer the target sentence. For instance, Luo et al. (2023) leverages the word-level edit-based prompt and designs a discrete searching algorithm to predict the target text. Chiyu et al. (2024) leverage LLM to generate reasoning paths and transferred texts for both parallel and non-parallel datasets, and use these generations to train a smaller task-specific model for TST.

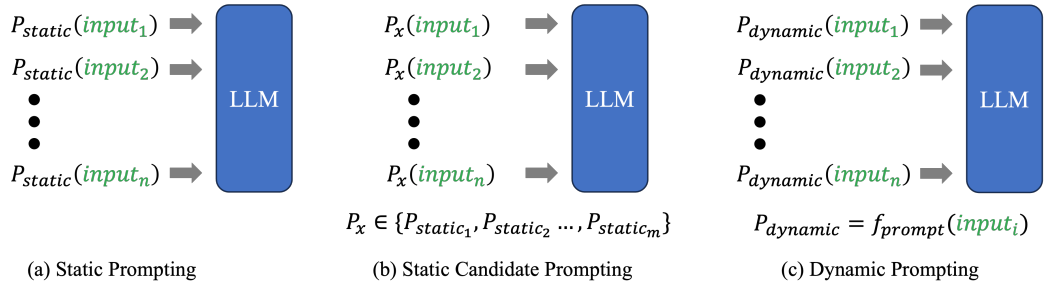


Figure 2.7: Three classes of prompt methods based on LLMs for the TST task.

We concluded all those prompting methods for the TST task as three paradigms outlined in Figure 2.7. The introduction for each paradigm is as follows:

- (1) **Static Prompting.** This is the simplest prompting paradigm. All the input sentences share the same template, P_{static} , in which each input is filled (Reif et al., 2022; Suzgun et al., 2022).
- (2) **Static Candidates.** To mine the better performance on diverse input cases, the

second approach designs a selection algorithm, P_x , to decide the best prompt template from a pre-designed set of candidates for each input (Liu et al., 2024).

- (3) **Dynamic Prompting** The last approach, dynamic prompting is more flexible and varies with each input. The prompt is output by a specific function, marked as f_{prompt} , which can be implemented with another LLM generating prompt (Chiyu et al., 2024) or a fixed procedure constructing prompt for each input.

2.2 Datasets and Benchmarks

High-quality datasets are crucial for establishing research benchmarks across various tasks. However, the construction of such datasets often requires specialized linguistic expertise. Although vast amounts of online data, such as SNS and forum comments or Wikipedia articles, are readily accessible, parallel corpora still heavily depend on manual annotation. To mitigate the issue, data augmentation techniques are widely used to extend pseudo-parallel datasets for specific tasks.

Due to the extensive diversity of text styles involved in TST, it is impractical to provide an exhaustive list of datasets employed for TST tasks. In this chapter, we will introduce three manually constructed datasets and two data augmentation methods tailored for TST.

2.2.1 Annotated Datasets

The traditional parallel dataset for TST is established by Rao and Tetreault (2018). They created 110k informal /formal sentence pairs derived from two domains, Entertainment & Music (E&M) and Family & Relationships (F&R), by using the Yahoo Answers L6 corpus¹.

The train set is annotated by leveraging the Amazon Mechanical Turk, a crowdsourcing tool², and finally evaluated by experts to delete incorrect annotation. To be of higher quality compared to the train set, they recruited a set of 85 human experts to annotate the test and validation sets. Based on this large parallel dataset, they set up a benchmark for research on FST tasks.

Compared with the FST task, it is more difficult to establish the parallel data for the SST task, because of the difficulty of defining a unified quantification standard to evaluate the sentiment conveyed in text. Although, for SST, collecting large-scale non-parallel datasets from the internet proves to be a more convenient approach, Li et al. (2018) utilized also hire crowdworkers on the Amazon Mechanical Turk tool to flip the

¹<https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&guccounter=1>

²<https://www.mturk.com/>

sentiment of each source sentence in the test set for both Yelp (Shen et al., 2017) and Amazon (He and McAuley, 2016) reviews. Subsequently, Suzgun et al. (2022) used a simple text-cleaning procedure to clean these two annotated test sets.

For SST, with a sentiment reversal also comes a reversal in meaning. To facilitate the research about the topic that neutralizes a negative point of view without contradicting the original meaning, Ziems et al. (2022) introduced a new subtask of TST, named positive reframing (or PTR). Simultaneously, they issued a parallel dataset to establish a benchmark for this task.

Instead of just showing the examples to crowd workers, Ziems et al. (2022) precisely define the sentiment and unify the annotation work by introducing six theoretically motivated reframing strategies to annotators. Moreover, the annotators are also required to attach the strategies in which they used. This design unifies the procedure of annotation to reduce the noise and errors caused by the preferences of each individual.

2.2.2 Data Augmentation

For the specific task, where it is difficult to obtain large annotated data, the pseudo-parallel datasets created by utilizing data augmentation techniques are ideal alternatives.

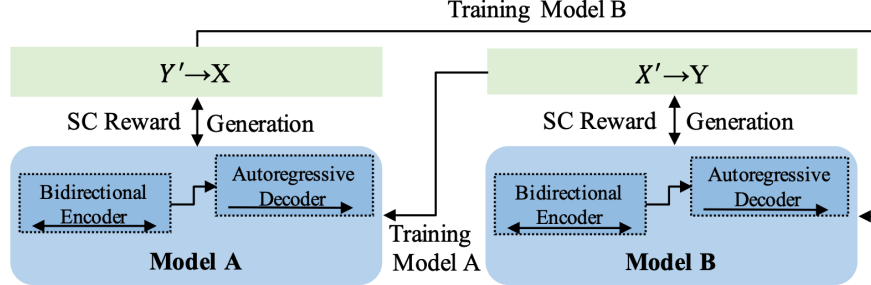


Figure 2.8: Iterative back translation for data augmentation (Lai et al., 2021a).

To extend the pseudo parallel data on the train sets for both Yelp and Amazon mentioned above, Lai et al. (2021a) leveraged the BART to set an iterative back translation framework illustrated in Figure 2.8. Given the non-paired X and Y with two opposite sentiment styles, the framework is fine-tuned by optimizing the content and style rewards in a reinforcement learning fashion until convergence. As a result, the generated X' and Y' are matched with X and Y , respectively to form two pseudo pairs, (X, X') and (Y, Y') .

In contrast, Liu and Chen (2022), directly uses the non-parallel dataset itself to search the possible pairs constrained by the lexical and semantic similarities. The dataset created using this method ensures that both sentences in a pair are sourced from the original dataset, rather than being generated by the PLM, while fewer pseudo pairs are available in the entire dataset.

2.3 Metrics for Evaluation

Evaluation serves as a critical role for assessing the effectiveness of various methods and as a guiding framework for exploring new approaches. In the context of TST-related research, three primary evaluation strategies are commonly utilized: human evaluation, automated evaluation, and evaluation leveraging LLMs. Furthermore, regardless of the evaluation approach utilized, the primary concern is to evaluate the model’s performance across three core dimensions: style transfer, content preservation, and fluency.

2.3.1 Human Evaluation

Intuitively, human evaluation is the most straightforward approach for assessing model performance. By defining scoring rules for the three dimensions mentioned above according to the requirements of a specific task, evaluators can accurately assess and compare various methods on benchmarks. To mitigate individual bias, multiple evaluators are often employed, and their scores are aggregated.

However, human evaluation presents two notable limitations. First, it is both time-intensive and expensive, as each work necessitates the creation of task-specific scoring criteria for manual assessment. Second, it is challenging to directly compare human evaluation results across different research, limiting its reusability and generalizability.

2.3.2 Automatic Evaluation

To overcome the shortcomings of human evaluation, most studies employ automatic evaluation methods. We outline commonly used metrics in previous works from the perspective of the three evaluation dimensions as follows.

- (1) **Style Transfer.** The most widely used metric is the accuracy of the generated outputs in accurately conveying the target style. The style category of the generated sentence can be predicted by a pre-trained style classifier or a third-party general style classifier.
- (2) **Content Preservation.** The similarity between prediction and original sentence is usually employed to evaluate the performance of content preservation. The similarity can be calculated by using different algorithms or models, such as BLEU score (Papineni et al., 2002), ROUGE score (Lin, 2004), and BERT score (Zhang et al., 2020a).
- (3) **Fluency.** To assess the fluency of the generated sentences, the primary automatic metric used is perplexity (Jelinek et al., 1977) or the average sentence length.

2.3.3 LLMs-based Evaluation

With the rapid development of LLMs. Straightforward prompting on LLMs for each dimension is also a promising automatic evaluation for TST. [Kocmi and Federmann \(2023\)](#) analyzed the correlation between human judgments and score output by ChatGPT. Experiments demonstrated that ChatGPT has the potential to evaluate the generations as an alternative to human evaluation.

In comparison to other automatic evaluation methods, the performance of LLM-based evaluation is highly contingent on the quality of the prompt. The design of task-specific prompts often depends on the expertise and experience of the researchers.

Chapter 3

Incorporating Sentiment Transfer and Paraphrase Generation for PTR

3.1 Motivation

PTR has been explored as one of the sub-fields in the TST study. Likewise, human-annotated data such as Positive Psychology Frames (PPF) has been created for this task by [Ziems et al. \(2022\)](#). According to its definition, given a negative claim or sentence, PTR involves generating text that conveys a positive perspective while preserving the inherent intention expressed by the input text. This task demands more sophisticated capabilities compared to traditional controllable text generation and has garnered significant attention as a prominent NLP research topic.

As outlined in section 1.2, one major approach for the TST task is to apply supervised learning for parallel data. Xu et al. ([Xu et al., 2019](#)) and Zhang et al. ([Zhang et al., 2020b](#)) attempted multi-task learning for parallel data. To mitigate the small size of the parallel data, Rao ([Rao and Tetreault, 2018](#)) presented data augmentation strategies. Another attempt is to utilize a non-parallel dataset and train a model in an unsupervised manner ([Shen et al., 2017](#); [Fu et al., 2018](#)). John et al. proposed a method that disentangles content- and style-related features and makes the decoder generate ideal output by using the disentangled features ([John et al., 2019](#)). Lai et al. designed two types of rewards for target style and content based on reinforcement learning ([Lai et al., 2021b](#)). Many of these methods attained significant progress on the TST task, while they still fail to handle the fine-grained generation, i.e., transfer the semantic-related style while preserving the intention of the input that is required for the PTR task.

Considering that reversing the sentiment attributes also entails reversing the intent of the text, most related works explored for TST and SST by fine-tuning PLMs can not be straightforwardly extended to PTR. Due to the significant representation capability

of the PLM, a beneficial baseline can be easily obtained by just fine-tuning the PLM. However, how to interpret a diversity of contexts to give a positive perspective is still an open problem. Especially, it is more serious when the training data is limited. Moreover, the absence of parallel datasets imposes additional constraints on the model’s generalization capabilities. The attempts training from scratch, based on disentangling or entangling methodologies and unsupervised learning, can generate transferred sentences by regularizing the hidden space, while they ignore the prior knowledge learned by the PLMs.

To mitigate these problems, we propose a PTR framework that automatically disentangles the meaning and style of text through two auxiliary tasks: paraphrase generation and sentiment transfer. Unlike traditional approaches that impose constraints on hidden states, our method decomposes the PTR task into these auxiliary tasks, facilitating a more structured learning process.

Furthermore, to further tackle the scarcity of parallel datasets for these auxiliary tasks, our approach leverages pseudo datasets generated using two data augmentation strategies. Then, a simple yet effective multi-task fine-tuning strategy is employed to integrate the generation capabilities derived from these datasets seamlessly.

The remainder of this chapter will elaborate on the specifics of our method and provide a detailed analysis of the experimental results.

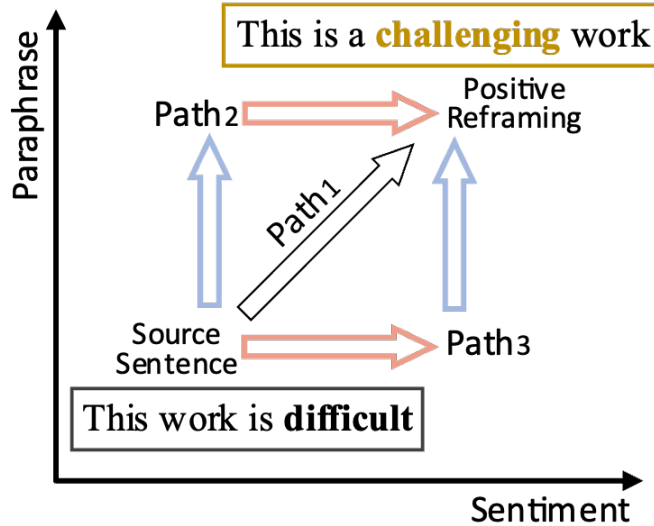


Figure 3.1: Decomposing PTR into Paraphrase Generation and Sentiment Transfer

3.2 Methodology

3.2.1 Decomposing PTR

The main challenge in the PTR task is how to control diversity and the extent of style transfer. The concept of decomposing PTR can be illustrated in Figure 3.1. The straightforward fine-tuning of PLM, proposed by (Ziems et al., 2022), is shown in the path, $Path_1$. We regard this strategy as our baseline which requires the model to directly learn the capability of paraphrase generation (PG) with diversity, and sentiment transfer (ST) with a positive perspective for the input.

However, it is challenging for the model to directly capture all of the complicated features at once. We thus divide this path into two components to make the problem easier i.e., PG and ST marked with blue and pink colors. Specifically, two paths $Path_2$ (from PG to ST) and $Path_3$ (from ST to PG) exist to obtain the target sentence. This method leverages two pseudo-datasets, paraphrase pairs with sentiment polarities, and sentiment pairs with paraphrases to disentangle meaning and style and transfer the source text into a diverse and positive target sentence.

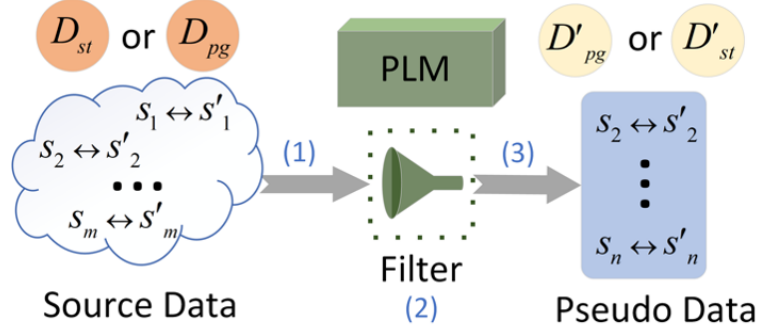


Figure 3.2: Filtering the original matched sentence pairs to obtain the pseudo dataset

3.2.2 Creating Pseudo Data as Prior Knowledge

(1) Selecting Annotation Pairs

We choose Microsoft Common Objects in COntext (MSCOCO) which are widely used to learn the paraphrase generation model. We call the data D_{pg} . Shen et al. modified the huge Yelp reviews dataset for sentence-level sentiment analysis (Shen et al., 2017). We utilized it to learn the sentiment transfer model. We divided it into two sets, S_{neg} and S_{pos} consisting of sentences with negative and positive sentiment labels, respectively. We created pairs for $\forall s_i \in S_{neg}$, and $\forall s'_i \in S_{pos}$. To reduce the computation cost, for a given s_i , we randomly chose the number of $0.05 \times |S_{pos}|$ samples from the set S_{pos} . We thus obtained a set D_{st} consisting of $0.05 \times |S_{pos}| \times |S_{neg}|$

sentence pairs.

(2) Filtering and Creating Two Pseudo Datasets

To create pseudo datasets from two datasets, D_{pq} and D_{st} , each sentence of a pair extracted from D_{pq} should be different polarity from each other. Similarly, each sentence of a pair from D_{st} should have a similar meaning. To this end, a semantic similarity classifier F_{sem} and a sentiment classifier F_{senti} are trained by leveraging BERT (Devlin et al., 2019). We utilized Semantic Textual Similarity Benchmark (STS-B) (Cer et al., 2017) and TweetEval Sentiment (TE-sentiment) (Barbieri et al., 2020) for training F_{sem} , and F_{senti} , respectively.

The semantic similarity score obtained by F_{sem} ranges from 0 to 5.0. The higher the score value, the more semantically similar the two sentences are. We chose sentence pairs from the set D_{st} whose similarity score δ is larger than a certain threshold value and obtained pseudo set D'_{st} . Likewise, we chose only two types of sentence pairs labeled with the fine-grained sentiment classifier F_{senti} , i.e., (Negative, Neutral) and (Neutral, Positive) from the set D_{pq} , resulting in pseudo set D'_{pg} . Therefore, we regard the construction of pseudo-parallel data as a unified filter procedure illustrated in Figure 3.2.

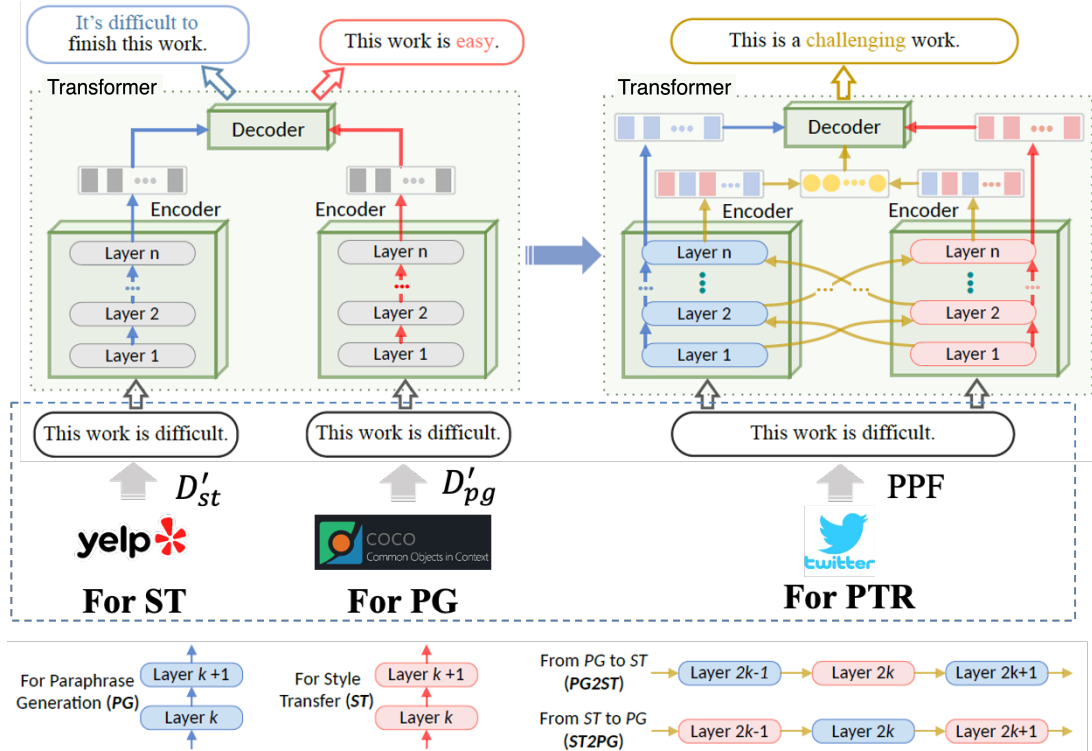


Figure 3.3: The model architecture and data flow: The architecture consists of two fine-tuning stages, **Stage 1** and **Stage 2**, and four data flows, PG , ST , $PG2ST$, and $ST2PG$.

3.2.3 Fusion Strategies

We recall that the straightforward fine-tuning of PLM illustrated in the path $Path_1$ of Figure 3.1 requires directly capturing all the complicated features at once. We thus divide this path into two relative steps: paraphrase generation, and sentiment transfer. The model architecture and data flow are illustrated in Figure 3.3. It consists of two fine-tuning stages and four data flows. More specifically, in stage 1, the PLM encoder is copied and each encoder is fine-tuned for PG and ST, respectively. We utilize the multi-task learning algorithm proposed by Liu et al. (Liu et al., 2019) to fine-tune the PLM on two pseudo datasets, D'_{pg} and D'_{st} . It can balance the PG and ST .

After processing stage 1, the same model is further fine-tuned on the PPF dataset following four data flow variants, PG , ST , $PG2ST$, and $ST2PG$. In stage 2, when the data flow is PG or ST , each independent encoder is utilized to fine-tune the model, while $PG2ST$ or $ST2PG$, both encoders are utilized. Let $E_{pg} = [l_{p_1}, \dots, l_{p_n}]$, and $E_{st} = [l_{s_1}, \dots, l_{s_n}]$ be the encoder for PG , and ST , respectively. Here, $l_{p_i} \in E_{pg}$ and $l_{s_i} \in E_{st}$ are the i -th block layer in the encoder (“Layer i ” of blue, and pink color in Figure 3.3, respectively). The encoder by $PG2ST$ and $ST2PG$ flows are shown as $E_{pg2st} = [l_{p_1}, l_{s_2}, \dots, l_{p_{n-1}}, l_{s_n}]$, and $E_{st2pg} = [l_{s_1}, l_{p_2}, \dots, l_{s_{n-1}}, l_{p_n}]$, respectively.

3.3 Experiments

3.3.1 Experimental Setup

We chose BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) pre-trained model as the PLM in our method (Lewis et al., 2020) since Ziems et al. (Ziems et al., 2022) reported that they provided the best quality of positive reframes among other PLMs such as GPT-2 (Radford et al., 2019) and CopyNMT (See et al., 2017). We utilized the version “facebook/bart-base”, and “t5-base” on Hugging Face¹ as the backbones. The statistics of datasets are summarised in Table 3.1. Semantic Textual Similarity Benchmark (STSB) (Cer et al., 2017) and TweetEval Sentiment (TE-sentiment) (Barbieri et al., 2020) are used to train the classifiers, F_{sem} , and F_{senti} , respectively.

We utilized the PPF dataset² to evaluate our method. It consists of 8,349 sentence pairs with manual annotation. The same BART trained in stage 1 is further trained on the PPF training set. The semantic similarity value δ is set to 3. We tuned the hyperparameters as follows: the batch size is 4, 8, 16, 32, the number of epochs is from 2 to 5, the number of layers n is 12, and the value of the learning rate is from $1e-5$ to $1e-4$. The procedure of tuning hyperparameters is automatically conducted by

¹<https://huggingface.co/models>

²<https://github.com/SALT-NLP/positive-frames>

the “Ray Tune”³ library.

For each variance, the detailed hyperparameters used for fine-tuning are outlined in the Appendix B. We implemented our model using PyTorch and accelerated the experiments with an NVIDIA GeForce RTX 3090ti GPU (24GB memory).

Dataset	Train	Validation	Test
PPF	6,679	835	835
D'_{pg}	15,181	134	1,899
D'_{st}	14,807	139	215
STSB	5,749	1,500	1,379
TE-sentiment	45,615	2,000	12,284

Table 3.1: The statistics of annotated dataset and pseudo dataset

For a fair comparison with the baseline by (Ziems et al., 2022), we used the eight metrics, which are (1) ROUGE-1, ROUGE-2, ROUGE-LCS (longest common subsequence) (Lin, 2004), BLUE (Papineni et al., 2002) and BERT-Score (Zhang et al., 2020a) referring to the gold reference for assessing the performance on content preservation and entire performance, (2) The Δ TextBlob value (Loria, 2018) for assessing the positivity transfer effectiveness, and (3) The Average Length and Perplexity (Yang et al., 2018b), followed by (Jin et al., 2022) for measuring the fluency of the output sentences.

3.3.2 Results

Table 3.2 shows the results on the PPF test dataset. We can see from Table 3.2 that the results obtained by our approach improve the performance compared with the baseline with the BART model except for BScore and Δ TB. Similarly, our results are better than the baseline with the T5 model except for Avg.Len. This shows that our approach contributes to giving a positive perspective while preserving the original contents. Our variants show that the BART is more effective than T5 by five metrics, R-1, 2, LCS, BLUE, and Avg.Len. However, the variants with T5 are more robust as they work well on content preservation (BScore), positivity transfer (Δ TB), and fluency (PPL).

The performance on the baseline by Avg.Len is more affected by the PLM model than our models as there is a significant difference (35.3-24.4) between T5 and BART baselines. Overall, $PG2ST$ and $ST2PG$ except for Avg.Len of T5, preserves the balance between the meaning of the contents and positivity as these results have medium scores between ST and PG .

Note that in the BART backbone, the results by the PG strategy are best on all

³<https://docs.ray.io/en/latest/tune/index.html>

Method	R-1	R-2	R-LCS	BLEU	BScore	Δ TB	Avg.Len	PPL
(Ziems et al., 2022)	27.7	10.8	24.3	10.3	89.3	0.23	24.4	-
ST (ours)	32.5	13.4	26.6	10.1	88.4	0.22	26.9	24.6
BART PG (ours)	32.8	13.7	27.1	10.6	88.3	0.17	26.8	26.6
PG2ST (ours)	32.6	13.5	26.9	10.3	88.4	0.19	26.7	24.8
ST2PG (ours)	32.9	13.6	27.1	10.9	88.4	0.20	26.6	25.6
(Ziems et al., 2022)	27.4	9.8	23.8	8.7	88.7	0.38	35.3	-
ST (ours)	31.1	11.2	25.4	8.9	88.7	0.39	24.3	14.0
T5 PG (ours)	30.8	11.2	25.5	8.7	88.7	0.33	23.5	15.4
PG2ST (ours)	31.1	11.2	25.5	8.9	88.7	0.35	23.4	14.5
ST2PG (ours)	30.8	11.3	25.5	8.8	88.7	0.33	23.0	15.1

Table 3.2: Main results Against the baseline (Ziems et al., 2022) on PPF dataset. ST and PG are the results obtained by only applying stage 1. R-1, R-2, and R-L refer to ROUGE-1, 2, and LCS. BSocre indicates BERT-Score and Avg.Len shows the Average length. The bold font indicates the best result obtained by each backbone.

ROUGE scores, while the *ST* strategy can perform best on average length and perplexity. The reason could be that for *PG*, the encoder is fine-tuned on D'_{pg} which is obtained from paraphrase generation data during the first stage in Figure 3.3. In contrast, the encoder used by *ST* is fine-tuned on D'_{st} whose source is sentiment data. Therefore, the model can perform better in terms of preserving the semantic features and sentiment transfer in *PG*, and *ST*, respectively. The *ST2PG* could balance the functions of *ST* and *PG* and obtain the best result on the BLEU score. Why the *PG2ST* can not perform similarly to *ST2PG* is still unknown and needs further investigation as future work.

For the backbone of T5, although the results obtained by our four strategies are better than those of the baseline except for the average length, the best performances by each metric are varied on all of these four variants without clear rules. We also need further investigation to make the reasons clear. To conclude our results, our two steps of fine-tuning combined with each strategy can provide different advantages on semantic preserving, sentiment transfer, and the balance of these two.

3.3.3 Case Study

Table 3.3 illustrates example sentences obtained by one of the variants of our approach, *ST2PG*, and the baseline with BART. We analyze the transferred examples by two variances with the ground truth, respectively. As shown in the sequences highlighted in blue and pink, the output sentences generated by our model express more positively than compared with the baseline, properly preserving the meaning of the given input. For instance, in sentence 1, the “*hope*” is a more positive expression and the rest part keeps the meaning and topic of the input sentence. In contrast, the output of the base-

Sentence 1	Why is it so hard to get one day of alone time, of peace and quiet???
Reference	I could really do with some peace and quiet for a day.
ST2PG	I hope I get one day of alone time, of peace and quiet .
Baseline	It's hard to get one day of alone time, of peace and quiet , but it's worth it when you get it.
Sentence 2	Its like Friday righ now with the stock market like wtf is going on Lmaoo.
Reference	It may be a confusing time, but I know that I can remain calm in the storm.
ST2PG	The stock market is going through a rough patch right now, but I'm sure it will get better .
Baseline	It's like Friday righ now with the stock market like wtf is going on Lmaoo.
Sentence 3	What's the opposite of sleeping like a baby? Because that's me as of late tired.
Reference	I have not been sleeping like a baby lately, but I will find a way to not be so tired.
ST2PG	I'm tired , but I'm sure I'll get better soon .
Baseline	I hope I get a good night's sleep so that I can get into a better frame of mind tomorrow.
Sentence 4	I WISH YOU WOULD JUST CHILL OUT! I am working as hard as I can.
Reference	I am doing the best I can rifht now.
ST2PG	I am working as hard as I can , but I know I can do it .
Baseline	I am working as hard as I can to make you happy.
Sentence 5	the reason why I'm wide awake right now is unexplainable and is gonna hurt me bad in the morning.
Reference	I don't know why I am not able to sleep, but I hope to get up fresh in the morning.
ST2PG	I'm wide awake right now, but I'm sure I'll feel better in the morn- ing.
Baseline	The reason why I'm wide awake right now is unexplainable and is going to hurt me bad in the morning, but I'm sure I can get through it.

Table 3.3: Transferred examples, The sequences marked with blue preserve the original contents, and those of pink give a positive perspective.

line is duplicated with the source input and appends a short sequence including three pronouns “it” which causes ambiguity, although the “it’s worth” is positive. In Sen-

tence 2, the baseline blindly copies the counterpart from the input, and just changes the word, “*Its*” to “*It’s*”. The output does not give positive perspectives, while it preserves the meaning of the original sentence. In contrast, “*is going through a rough patch*” which is generated by our model *ST2PG* shows that it keeps the bad situation of the stock market, and proposes a positive wish by using the sentence, “*I’m sure it will get better*”.

Likewise, in sentences 3 and 4, the output from our *ST2PG* can balance the content preserving and sentiment transfer, while the baseline only captures either content preserving or sentiment transfer. For the special case, in sentence 5, “*I’m sure I can get through it*” in the baseline is positive, while its meaning is too abstract and the context is not suitable. In contrast, the output of our *ST2PG*, “*I’ll feel better in the morning*”, is very similar to “*get up fresh in the morning.*” in the ground truth.

3.4 Summary

In this work, we proposed a method for learning disentangled meaning and style representations by fine-tuning PLMs on two auxiliary tasks for PTR. Our approach utilizes pseudo-positive reframing datasets and a simple multi-task learning-based model is learned to fuse the generation capabilities from these datasets. The results by using the PPF dataset showed the efficacy of our approach.

3.4.1 Contribution

The contributions of this paper can be briefly summarized as three aspects: (1) we propose a simple but effective reframing model for the PTR task, (2) we propose two data augmentation strategies for generating pseudo-positive reframing datasets, and (3) The experimental results on PPF dataset, show that our approach outperforms the baselines, BART by five and T5 by six evaluation metrics. To rebuild the experiments and facilitate further work, our source codes and data are available online ⁴.

3.4.2 Limitation

The limitations of this work can be divided into two primary aspects: dataset constraints and model fine-tuning.

While auxiliary tasks can simplify PTR by breaking it into multiple steps, the lack of parallel datasets necessitates the additional effort of constructing new datasets. Naturally, the quality of these datasets directly affects the model’s performance. Additionally, the increased data requirements lead to higher computational costs.

⁴<https://github.com/codesedoc/DMSR>

Furthermore, the multi-task training approach relies on a critical assumption: that for PG, ST, and PTR, all sentences share a common latent space. Based on this assumption, the model is expected to utilize the knowledge gained from the auxiliary tasks to enhance its performance on PTR.

3.4.3 Future Work

Considering the limitations discussed above, a promising direction for future work is to develop more efficient fine-tuning architectures that eliminate the need to rely on additional datasets. To this end, the next chapter introduces an alternative method that leverages only the available parallel PPF dataset to regularize the hidden states of PLMs for PTR.

Furthermore, other valuable directions for exploration include: (1) exploring more effective augmentation strategies by leveraging a huge number of unlabeled datasets, (2) evaluating our method by using other PPF datasets, and (3) applying it to other TST tasks such as formality and politeness.

Chapter 4

Disentangling Style from Content by Contrastive Learning for PTR

4.1 Motivation

As discussed in Chapter 3, for the PTR task, large annotated pairs of datasets are not available and would be expensive and time-consuming to create. Therefore, how to interpret a diversity of contexts and generate a positive perspective from a small size of the training dataset is still an open problem.

The PTR task requires both the Sentiment Transfer (ST) and Paraphrase Generation (PG) tasks, i.e., transferring the sentiment of the source text, and more on the linguistic diversity of generation while preserving the meaning of the source text. The PTR is a sub-branch of Sentiment Style Transfer (SST) with sentiment polarity. As mentioned by [Ziems et al. \(2022\)](#), the target of PTR is to neutralize the given negative point of view and generate a more positive perspective without contradicting the original meaning, while the SST takes care more of the polarity of the sentence, and reverse the sense of source sentence with reversing its sentiment. The sentiment is a content-related attribute of text, and the requirement of transferring the sentiment of the given input for the PTR task is a fine-grained transfer.

To establish the benchmark, [Ziems et al. \(2022\)](#) experimented with many of the leading PLMs including BART ([Lewis et al., 2020](#)), and T5 ([Raffel et al., 2020](#)), i.e., directly fine-tune these models on their collected PPF parallel dataset, and showed that these models can learn to shift from a negative to a more positive perspective. However, as reported by them, these models still struggle to generate reasonable positive perspectives. Many previous attempts on the TST task and even less work on the PTR task are either too complex or unable to handle subtle transitions in style. As a result, they are still limited in generating diverse contexts.

In this work, we propose a simple but effective **Framework to Decouple Style from**

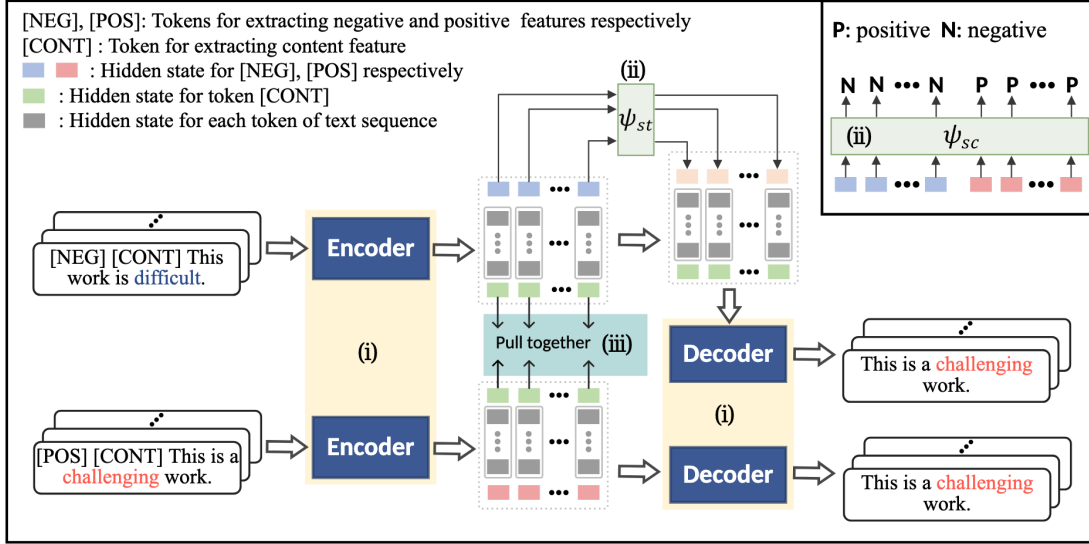


Figure 4.1: The architecture and data flow of FDSC which consists of (i) sequence-to-sequence text generation, (ii) decomposing style from contents, and (iii) preserving primary content.

Contents (FDSC) for the PTR task. Different from the previous work on the PTR task that utilizes PLM to directly fine-tune the task-specific labeled dataset, PPF, our method decomposes an input sentence into the sentiment attribute and primary content that refers to the factual information and transfers the sentiment feature in the hidden space while preserving the semantic content of the input sentence.

Our motivation for such a strategy is based on the assumption that the PTR is relatively more fine-grained than the common SST and other TST tasks. To this end, we leverage contrastive learning to disentangle sentiment features from the content and learn more fine-grained and robust representations of sequences for preserving semantic meaning.

4.2 Method

Given a sequence/sentence with a negative sentiment, the goal of PTR is to reframe the source input into a target sequence/sentence with a relatively positive perspective while preserving the original sense of the given input. For this task, we propose an approach, FDSC, by leveraging PLM as the backbone model. As illustrated in Figure. 4.1, FDSC consists of three modules, (i) sequence-to-sequence (Seq2Seq) text generation, (ii) decoupling style from contents, and (iii) preserving primary content. These components are simultaneously trained by adopting loss functions.

4.2.1 Seq2Seq Text Generation

Ziems et al. (Ziems et al., 2022) mentioned that the base requirement for PTR is that the model is diverse and fluent in reframing while preserving the original sense of the given sentence with positive perspectives. Following their work, we propose a simple but effective PTR approach, fine-tuning the pre-trained transformers for annotated data. As shown in (i) of Figure. 4.1, we choose transformers consisting of an encoder-decoder structure as the backbone of FDSC and directly control the model to decouple sentiment from the primary content of the input sequence. Let \mathbf{x} be a source sequence and \mathbf{y} be its annotated target, the parallel dataset with N samples can be formulated as $D = \{(\mathbf{x}_i, \mathbf{y}_i) | 1 \leq i \leq N, i \in \mathbb{Z}\}$. The set of all \mathbf{x} , and \mathbf{y} in D can be labeled as X and Y respectively, therefore, the population of sequences in the dataset is represented as $S = X \cup Y$. Let the prediction of the transformer referring to $\mathcal{G}_{seq2seq}(\mathbf{s})$, $\mathbf{s} \in S$. The cross entropy loss between \mathbf{y} and $\mathcal{G}_{seq2seq}(\mathbf{x})$ is given by $f_{ce}[\mathcal{G}_{seq2seq}(\mathbf{x}), \mathbf{y}]$. The objective of the backbone during fine-tuning is to minimize the following training loss:

$$\mathcal{L}_{neg2pos} = -\frac{1}{N} \sum_{i=1}^N f_{ce}[\mathcal{G}_{seq2seq}(\mathbf{x}_i), \mathbf{y}_i], \quad (\mathbf{x}_i, \mathbf{y}_i) \in D. \quad (4.1)$$

Different from the bidirectional transfer target of SST, the transfer of sentiment attribute is unidirectional in PTR, which exclusively changes the polarity of the sentence from negative to relatively positive. To make the model robust, and leverage the manual annotated parallel corpus, for each target reference \mathbf{y}_i of each \mathbf{x}_i , \mathbf{y}_i is also pushed into the backbone of FDSC and reconstructs itself which is similar to the auto-encoding learning strategy. The training loss is given by:

$$\mathcal{L}_{auto} = -\frac{1}{N} \sum_{i=1}^N f_{ce}[\mathcal{G}_{seq2seq}(\mathbf{y}_i), \mathbf{y}_i], \quad \mathbf{y}_i \in Y. \quad (4.2)$$

The final loss obtained by the transformers is given by:

$$\mathcal{L}_{seq2seq} = \mathcal{L}_{neg2pos} + \mathcal{L}_{auto}. \quad (4.3)$$

4.2.2 Decoupling Style from Contents

To model diverse and fluent reframing, our FDSC fine-tunes the model for the input sequence with two special symbols to decouple style from the contents. Each input sequence, $\mathbf{s} \in S$, is pre-fixed to two special tokens. One is marked as t_s^s which shows the polarity of the sentiment of the input with [NEG] (negative) or [POS] (positive). Another is t_c^s which indicates the sequence-level primary content with [CONT].

Let $(t_1^s, t_2^s, \dots, t_n^s)$ be the tokenized sequence of \mathbf{s} with the length of n . We create the entire input sequence of the model, $\mathbf{t}^s = (t_s^s, t_c^s, t_1^s, t_2^s, \dots, t_n^s)$. For a given \mathbf{t}^s , the hidden

state output from the encoder is indicated as $\mathbf{H}^s = (\mathbf{h}_s^s, \mathbf{h}_c^s, \mathbf{h}_1^s, \mathbf{h}_2^s, \dots, \mathbf{h}_n^s) \in \mathbb{R}^{n \times d}$, where d denotes the dimensions for each token. We utilized $\psi_{sc}: \mathbb{R}^d \rightarrow (0, 1)$ which is illustrated in (ii) of the right-hand side of Figure. 4.1 so that \mathbf{h}_s^s is capable to represent the sentiment feature. Let $\psi_{sc}(\mathbf{s}) \in (0, 1)$ be the probability of positive polarity, and the binary cross entropy is chosen as the loss function. Thus, the training objective for decoupling sentiment style is to minimize the following equation:

$$\mathcal{L}_{sc} = -\frac{1}{2N} \sum_{i=1}^{2N} \{ \mathbb{1}(\mathbf{s}) \cdot \log[\psi_{sc}(\mathbf{s})] + [1 - \mathbb{1}(\mathbf{s})] \cdot \log[(1 - \psi_{sc}(\mathbf{s}))] \}, \mathbf{s} \in S, \quad (4.4)$$

$$\mathbb{1}(\mathbf{s}) = \begin{cases} 0, & \text{if } \mathbf{s} \in X, \\ 1, & \text{else if } \mathbf{s} \in Y. \end{cases}$$

Assume that the hidden states of the sentiment which belong to the same polarity group are close to each other in the hidden space. We thus add a sentiment transfer module $\psi_{st}: \mathbb{R}^d \rightarrow \mathbb{R}^d$ to FDSC to transfer the negative polarity of the input into the positive one which is illustrated in (ii) of the left-hand side of Figure. 4.1. The loss for the sentiment style transfer is given by:

$$\mathcal{L}_{st} = -\frac{1}{N} \sum_{i=1}^N |\psi_{st}(\mathbf{h}_s^{\mathbf{x}_i}) - \mathbf{h}_s^{\mathbf{y}_i}|^2, \quad (\mathbf{x}_i, \mathbf{y}_i) \in D, \quad (4.5)$$

where $\mathbf{h}_s^{\mathbf{x}_i}$ refers to the hidden state of the sentiment token in \mathbf{x}_i , and $\mathbf{h}_s^{\mathbf{y}_i}$ indicates the hidden state of the counterpart in \mathbf{y}_i . The final loss for decoupling style from the content and transferring it to positive polarity can be given by:

$$\mathcal{L}_{dsc} = \mathcal{L}_{sc} + \mathcal{L}_{st}. \quad (4.6)$$

4.2.3 Preserving Primary Content

The primary content shares the meaning of the factual information obtained by each pair of \mathbf{x} and \mathbf{y} . Therefore, their representations, $\mathbf{h}_c^{\mathbf{x}}$ and $\mathbf{h}_c^{\mathbf{y}}$, should be close to each other in the hidden space. Let $B_x = \{\mathbf{h}_c^{\mathbf{x}_1}, \mathbf{h}_c^{\mathbf{x}_2}, \dots, \mathbf{h}_c^{\mathbf{x}_b}\}$ be a mini-batch with the size of b during fine-tuning. Every two hidden states from an arbitrary two input sequences, $\mathbf{h}_c^{\mathbf{x}_i}$ and $\mathbf{h}_c^{\mathbf{x}_j}$ ($i \neq j$) should be apart from each other as they obviously represent different content. Likewise, the parallel batch $B_y = \{\mathbf{h}_c^{\mathbf{y}_1}, \mathbf{h}_c^{\mathbf{y}_2}, \dots, \mathbf{h}_c^{\mathbf{y}_b}\}$ consisting of the counterpart of each element of \mathbf{x} in B_x has the same manner. To enhance the model to learn a more robust contextual representation and generate the output sequence while preserving the original sense of the given input sequence, we apply the CL strategy which is illustrated in (iii) of Fig4.1.

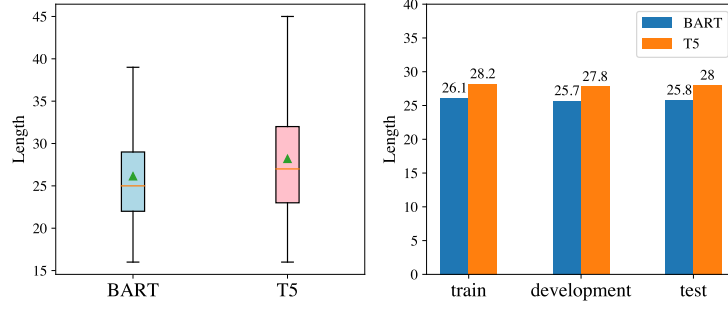


Figure 4.2: The statistics of the length of tokenized sequences on the PPF dataset. The right-hand side plot illustrates the average sequence length and the left-hand side shows the length distributions obtained by T5 and BART, respectively.

Following the popular version of contrastive loss proposed by Chen et al. (Chen et al., 2020), we also leverage the NT-Xent (the normalized temperature-scaled cross entropy) loss as the training objective. For $B = B_x \cup B_y$, the loss is given by:

$$\mathcal{L}_{contr} = -\frac{1}{2b} \sum_{i=1}^{2b} \log \frac{\exp[\phi(\mathbf{h}_c^{\mathbf{x}_i}, h_c^{\mathbf{y}_i})]/\tau}{\sum_{\mathbf{h}_c \in B_i} \exp[\phi(\mathbf{h}_c^{\mathbf{x}_i}, \mathbf{h}_c)]/\tau} \quad (4.7)$$

$$B_i = B - \{\mathbf{h}_c^{\mathbf{x}_i}, \mathbf{h}_c^{\mathbf{y}_i}\}, \quad \phi(\mathbf{u}, \mathbf{v}) = \mathbf{u} \cdot \mathbf{v} / \|\mathbf{u}\| \|\mathbf{v}\|$$

where $\phi(\mathbf{u}, \mathbf{v})$ refers to the cosine similarity function between two embedding \mathbf{u} and \mathbf{v} , and τ is a temperature hyper-parameter. The entire training loss of our FDSC is given by:

$$\mathcal{L} = \mathcal{L}_{seq2seq} + \mathcal{L}_{dsc} + \mathcal{L}_{contr} \quad (4.8)$$

4.3 Experiments

4.3.1 Experimental Setup

We constructed the experimental setup using the same backbone PLMs (i.e., BART and T5) and the PPF dataset as utilized in Chapter 3. To compare the tokenization effects on the same PPF dataset, we performed a statistical analysis of the sentence lengths after preprocessing with their respective tokenizers. Figure 4.2 shows that the average length obtained by T5 is approximately 25, which is slightly larger than that of BART.

We optimized the same three hyperparameters, batch size, number of epochs, and learning rate, by using the development set and the “Ray Tune” library. The specific values of these hyperparameters for each variant in the ablation study are provided in Appendix B. The size of the hidden layer of the sentiment transfer, ψ_{st} , in the FDSC, is

set to 512, and other hyper-parameters are consistent with the baseline and the default setting of the transformer package from Hugging Face.

Considering the motivation is to attempt the effectiveness of FDSC by comparing it with the straightforwardly fine-tuned baseline, we choose the same eight metrics we used in Chapter 3 to evaluate each model. These metrics include five items for assessing the entire performance and the capability of preserving content (ROUGE-1, ROUGE-2, ROUGE-LCS, BLEU, and BERT-Score), one item for assessing the performance on content preservation ($\Delta\text{TextBlob}$) value, and two items for measuring the fluency of the output sentences (Average Length and Perplexity).

4.3.2 Results

Table 4.1 shows the results of our FDSC against the baseline which directly fine-tuned the PLMs. The performances of FDSC by R-1, R-2, and R-CLS are better than the baseline by both of the PLMs, BART and T5, especially, our FDSC with BART and T5 by R-1 metric attained 18.1%, and 10.9% improvements, respectively. This indicates that FDSC contributes to preserving the meaning of the input. For BLEU, BScore, and ΔTB metrics, both FDSC and the baseline models have similar performance. Although the BScore obtained by FDSC with BART, and BLEU by FCSC with T5 are slightly worse than the baseline, there are no significant differences between them as both gaps are less than 0.8%.

Ziems et al. (2022) reported that the Avg.Len obtained by T5 gains good performance compared with other models such as 21.1 tokens by GPT-2 and 16.1 tokens by CopyNMT as the length by T5 is longer than 35 tokens. Our FDSC has a similar tendency, i.e., the Avg.Len obtained by our model with BART achieves good performance compared with the baseline since the generated sequence by our model is 5.4 tokens longer than the baseline. The improvements obtained by our method based on BART are better than those of the T5 backbone for 5 metrics, i.e., R-1, R-2, R-LCS, BLEU, and Avg.Len. It shows that our method is suitable for BART compared with T5. One of the possible reasons is that the BART used both encoder and decoder in the pre-training phase can learn more latent patterns correlated with those potentially affecting the PTR task.

4.3.3 Ablation Study

We conducted an ablation study to examine the effects of each module of our FDSC. The result is shown in Table 4.2. We can see that the model without decoupling style from the content (w/o decoup) can perform almost best on R-1, R-2, R-LCS, and BLEU, while the worst on BScore, ΔTB , Avg.Len and PPL. This indicates that the

Method		R-1	R-2	R-LCS	BLEU	BScore	Δ TB	Avg.Len	PPL
BART	Baseline	27.7	10.8	24.3	10.3	89.3	0.23	24.4	-
	FDSC	32.7	13.4	27.0	10.4	88.5	0.21	29.8	24.0
T5	Baseline	27.4	9.8	23.8	8.7	88.7	0.38	35.3	-
	FDSC	30.4	10.9	25.1	8.1	88.8	0.39	25.0	13.1

Table 4.1: Main results obtained by our FDSC against the baseline, which directly fine-tuned the PLMs(Ziems et al., 2022) on the PPF dataset. R-1, R-2, and R-L refer to ROUGE-1, ROUGE-2, and ROUGE-LCS. BScore indicates BERT-Score and Avg.Len shows the Average length. The **bold** font indicates the better result obtained by each backbone.

Method		R-1	R-2	R-LCS	BLEU	BScore	Δ TB	Avg.Len	PPL
BART	FDSC	32.7	13.4	27.0	10.4	88.5	0.21	29.8	24.0
	w/o decoup	32.7	13.9	27.3	11.0	88.1	0.10	26.2	28.6
	w/o contr	32.9	13.4	27.0	10.5	88.4	0.19	27.5	24.2
T5	FDSC	30.4	10.9	25.1	8.1	88.8	0.39	25.0	13.1
	w/o decoup	30.6	11.1	25.4	8.7	88.8	0.37	24.3	14.0
	w/o contr	29.9	10.6	24.7	8.3	88.7	0.39	24.7	13.4

Table 4.2: Ablation study of our FDSC. “w/o decoup” refers to removing the module on decoupling style from the content (training loss \mathcal{L}_{dsc} in Eq. (4.8)), and “w/o contr” indicates removing the contrastive learning module employed for preserving invariant contents (training loss \mathcal{L}_{contr} in Eq. (4.8)). The **bold** font shows the best result in each group.

component for contrastive learning can contribute to preserving the meaning of source sentences. In contrast, the model without contrastive learning (w/o contr) obtains generally better results on Δ TB, Avg.Len and PPL but the worse result on R-1, R-2, and R-LCS compared with the variant “w/o decoup”.

Avg.Len obtained by FDSC is better than those obtained by the other two models on both BART and T5, especially the improvement compared with the model without decoupling style from the content (w/o decoup) by BART and T5 are 13.7% and 2.9%, respectively. Likewise, the improvement of PPL obtained by FDSC works well, especially the improvement compared with the model without decoupling style from the content (w/o decoup) with BART and T5 are 19.2% and 7.6% respectively. Our FDSC works well on BScore, Δ TB, Avg.Len. PPL is relatively middle results by the other matrices. From these observations, we can conclude that the module of decoupling style from the content contributes to the fluency of output sentences and strikes a balance between style transfer and meaning preservation.

Sentence 1	“So exhausted. :(Only so much I can do in all areas of my life in a day/week but it never ends! :(frustrated.”
Reference	I have a lot to do, but I’m really proud of myself for keeping it all together.
FDSC	I have a lot to do in a day/week , but I know I can do it .
Baseline	I have so much to do in all areas of my life in a day/week , but it never ends. I hope to finish it soon.
Sentence 2	It is like Friday right now with the stock market like wtf is going on Lmaoo.
Reference	It may be a confusing time, but I know that I can remain calm in the storm.
FDSC	I hope the stock market improves soon so that I can get a good night’s sleep. The stock market is going through a rough patch right now, but I’m sure it will get better .
Baseline	It’s like Friday right now with the stock market like wtf is going on Lmao.

Table 4.3: Examples from the test set of PPF, their ground truth, and generated sentences by FDSC and Baseline. The sequences marked with blue preserve the original contents, and those marked with pink give a more positive perspective compared with the given input sentences.

4.3.4 Case Study

To more directly assess the effectiveness of our method, we selected two transfer cases and compared the sentences generated by the FDSC and baseline methods for the same input. Table 4.3 illustrates these example sentences, their ground truth reference, outputs generated by our FDSC with BART, and the baseline. As shown in the sequences highlighted in blue and pink color, the output sentences generated by our model express more positively compared with the baseline, properly preserving the meaning of the given input.

For instance, in Sentence 1, we can see that the baseline blindly copies the counterpart from the input, and just removes the words, “can” because the sequence, “do in all areas of my life in a day/week” is the same expression. It preserves the meaning of the original sentences while the output gives a weak positive perspective compared with that of FDSC. In contrast, “I have a lot to do in a day/week” which is generated by FDSC shows that it keeps the negative situation implicitly while giving a positive wish by using the sentence, “but I know I can do it.”

It is interesting to note that in Sentence 2, the output obtained by the baseline is the same as the input sentence, just changing the word, “It is” to “It’s”. It also can preserve the meaning of the original sentences while having less diversity and fewer positive

perspectives. In contrast, the output obtained by FDSC comprises long sentences. It gives positive perspectives by generating words/phrases, “*hope*”, “*improves soo*”, and “*it will get better*”, while it preserves the primary content, “*the stock market like wtf is going on Lmaoo*” by utilizing diverse expressions, “*The stock market is going through a rough patch.*”

4.4 Summary

To explore the fine-tuning performance of PLMs on limited labeled data, we proposed a simple but effective text reframing framework, FDSC by leveraging the Pre-trained Language Models as a backbone to decouple the sentiment style from the text content for the PTR task. The experimental results demonstrate the efficacy of our proposed approach, especially the ablation study shows that the module of decoupling style from the content contributes to the fluency of output sentences. Our code is available on GitHub¹.

4.4.1 Contribution

Compared with fine-tuning PLMs directly or training the entire model with reinforcement learning as in previous studies, we leverage contrastive learning to develop a novel training framework that imposes constraints on the model’s latent space, thereby extending the applicability of PLMs to TST tasks. The contributions of our work can be summarized:

- (1) We propose a simple but effective PTR framework to decompose an input sentence into the sentiment style and primary content, instead of directly fine-tuning the model.
- (2) We utilize contrastive learning to enhance the model to learn a more fine-grained and robust contextual representation to preserve the original meaning of the given input sentence.

4.4.2 Limitation

Although this work introduces a fine-tuning strategy that eliminates the need for auxiliary tasks and pseudo datasets compared to the approach discussed in Chapter 3, FDSC remains applicable only to downstream tasks with parallel datasets. For other TST tasks or domain-specific applications, the generalizability of fine-tuning-based methods remains inconsistent.

¹<https://github.com/codesedoc/FDSC>

On the other hand, the proposed method is based on a fundamental assumption: the hidden space produced by the Transformer encoder can be effectively constrained during fine-tuning through the introduction of specifically designed loss functions. However, this training objective is not integrated into the pretraining phase. Additionally, similar to the model introduced in Chapter 3, FDSC is limited to PLMs with an encoder-decoder architecture, such as BART and T5, and is incompatible with GPT-series models.

4.4.3 Future Work

The emergence of LLMs has enabled them to achieve performance comparable to, or even surpassing, that of fine-tuning PLMs on parallel datasets in downstream tasks. This is particularly advantageous in scenarios where parallel data is unavailable. Due to the absence of parameter updates, LLM-based prompting presents a promising direction as it reduces reliance on labeled datasets and operates under more flexible assumptions. In the next chapter, we will introduce a prompting strategy to transfer the sentiment style of input sentences.

Additionally, there are several other valuable avenues for future research, for instance, (1) exploring effective augmentation strategies by leveraging a large volume of unlabeled datasets, and (2) extending our method to other TST tasks, such as formality and politeness.

Chapter 5

Rethinking SST as Reduction and Synthesis

In Chapters 3 and 4, we introduced two fine-tuning approaches for the PTR task, both of which heavily rely on parallel data, limiting their applicability. This chapter explores an LLM-based prompting method and validates its effectiveness under few-shot conditions (without reliance on labeled data) on another sentiment-related TST subtask, i.e., the SST task.

5.1 Motivation

SST, as a variant of TST, has recently attracted extensive interest. Some disentangling-based approaches have improved performance, while most still struggle to properly transfer the input as the sentiment style is intertwined with the content of the text, similar to the challenges encountered in PTR tasks. To alleviate the issue, we propose a plug-and-play method that leverages an iterative self-refinement algorithm with a LLM.

As demonstrated in the previous Chapters, the disentanglement, i.e., disentangling style from text then fusing target style in hidden space corresponding to domain-specific data, has been indeed repeatedly proven to be a feasible approach (Shen et al., 2017; John et al., 2019; Bao et al., 2019; Lee et al., 2021; Sheng et al., 2023; Hu et al., 2023). However, these disentanglement-based approaches still suffer from mainly two insufficiencies.

- (1) It is not clearly shown that the semantic representation is entirely disentangled from the original style representation (Lee et al., 2021). Especially, Jin et al. (2022) demonstrated the sentiment style, unlike formality features, is more of a content-related attribute. For example, in transforming the negative input “I

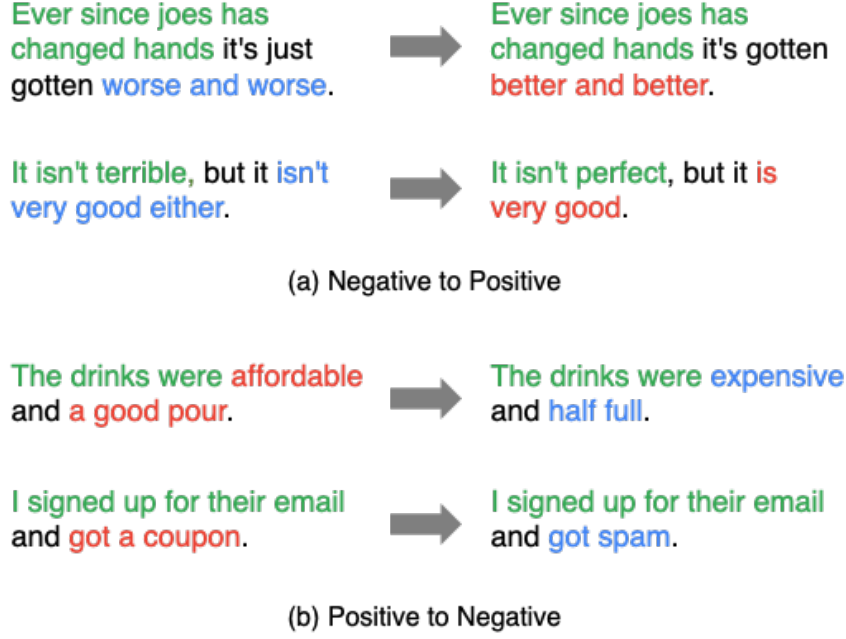


Figure 5.1: Examples of SST: (a) from negative to positive and (b) from positive to negative. The words with green color refer to the style-free content, and the blue and red fonts indicate the parts with negative and positive styles in context, respectively.

hate making decisions” into the positive output “*I love making decisions*”, the semantics would reverse along with the sentiment style (Ziems et al., 2022).

- (2) Few works address the issue that the challenging case is variable among the transfer cases. For example, as shown in (a) of Figure 5.1, it is easy to transfer from “*Ever since Joe has changed hands it’s just gotten worse and worse.*” to “*Ever since Joe has changed hands it’s gotten better and better.*”. However, it is difficult to transfer from “*It isn’t terrible, but it isn’t very good either.*” to “*It isn’t perfect, but it is very good.*”. The reason is that the sentiment style of the input, i.e., “*isn’t terrible*” (neutral) and “*isn’t very good*” (negative) is intertwined with the content of the sentence.

In Chapter 2, we reviewed some popular prompt-based methodologies that have been extensively studied and have obtained outstanding performances, especially by leveraging large language models (LLMs). (Reif et al., 2022; Suzgun et al., 2022; Luo et al., 2023; Liu et al., 2024). These efforts have achieved significant progress in the TST task; however, they have yet to fully explore the potential of LLMs to explicitly disentangle content-related style, such as sentiment polarity, from the given text in SST.

To this end, in this Chapter, we present a simple, yet effective plug-and-play method for the relatively challenging cases in a specific SST task by leveraging the LLM augmented with the Self-Refine algorithm (Madaan et al., 2023). We define that, for a

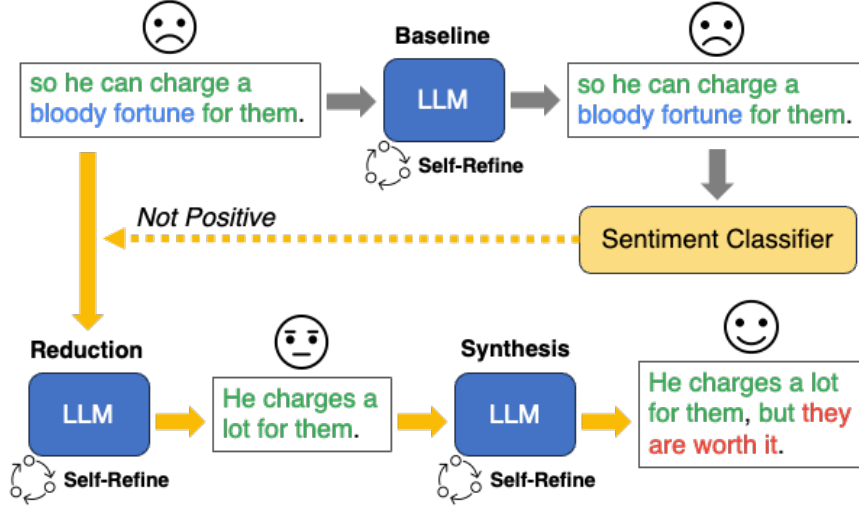


Figure 5.2: The pipeline of the reduction-synthesis method by leveraging LLM and Self-Refine algorithm. The words with green color express the style-free content, and the blue and red fonts indicate the parts with negative and positive styles respectively.

specific model and SST task, the samples in the dataset that can not be transferred accurately are more challenging cases. To address such SST cases, our plug-and-play method generates the target style via two phases, i.e., reduction and synthesis, which lead to LLM mining style-free sequence from the input text, and re-generate the target text by adding the target style to the style-free sequence.

Unlike the “disentangling” and “aggregating” operations applied to hidden states, the reduction and synthesis phases ensure that the model can differentiate between sentiment and style-independent content within a sentence, expressed directly in natural language.

5.2 Methodology

5.3 Plug-and-Play Approach

Figure 5.2 shows our straightforward plug-and-play method by illustrating an example of a challenging case from the Yelp dataset for transferring the negative to the positive style. We first apply the sentiment classifier to the output of the baseline model and detect the challenging cases, i.e., the sentiments of the generations obtained by the baseline model are incorrect. We then use our plug-and-play method to transfer these cases instead of the baseline.

As illustrated in Figure 5.2, the baseline just duplicates the input text with negative sentiment, “so he can charge a bloody fortune for them.”. In contrast, our plug-and-play method deals with the input in the first phase, **Reduction**, to detect a style-free

sequence, “He charges a lot for them.”. The output is then passed to the second phase, **Synthesis**, to generate the expected positive output: “He charges a lot for them, but they are worth it.”. To do this, we formulate the SST task and further decompose the SST into two sub-objectives with lower boundaries.

5.3.1 Problem Formulation

Let D be a set of text. Each sequence in D contains a sentiment style, *positive* (pos), *negative* (neg), or *neutral* (neu). For the SST task, we considered two main transfer cases i.e., from *positive* to *negative* and from *negative* to *positive* ($pos \rightleftharpoons neg$). Given a pair of source text X , and its target counterpart Y with a sentiment style label s , e.g. *positive*, the objective of the SST task can be formulated as the language model $\mathbb{P}(Y|X, s)$, where $s \in \{pos, neg\}$ and $X, Y \in D$.

Let also C be a style-free content text. We assume that one such neutral text C which should be preserved during transferring from X to Y exists. The objective of SST can be further decomposed as follows:

$$\mathbb{P}(Y|X, s) = \underbrace{\mathbb{P}(C|X)}_{\text{reduction}} \underbrace{\mathbb{P}(Y|X, C, s)}_{\text{synthesis}} \quad (5.1)$$

The detailed derivation of Eq. (5.1) is shown in the Appendix A. Following the derivation in Eq. (5.1), the optimization of the objective of the SST task can be decomposed into two components, reduction and synthesis, with lower boundaries.

5.3.2 Reduction and Synthesis

Note that the autoregressive pre-trained objective is more inherently similar to the optimization components of Eq. (5.1) and has outstanding performance for open-end text generation. We thus prompt the LLM to infer a proper style-free content C from X . We call this procedure as reduction phase. We then lead the model to generate the expected target by another prompt, called as synthesis phase. Inspired by Kojima et al. (2022), the reduction and synthesis (**RS**) can be regarded as a guidance that helps the pre-trained language model to transfer the sentiment polarity of the source sequence along with a chain-of-thought (CoT).

Moreover, for each phase, we leverage the Self-Refine algorithm, which is a specific resolution to mitigate the common hallucination issues and is often used in LLMs-based systems.

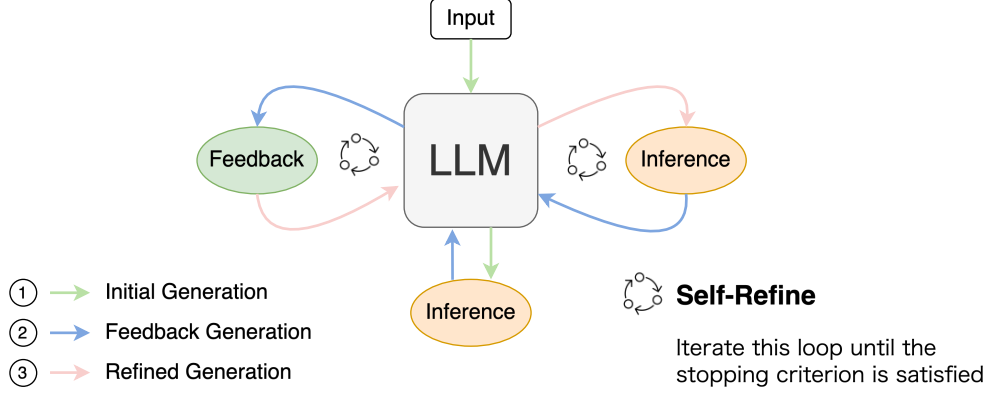


Figure 5.3: The framework of self-refine (Madaan et al., 2023)

5.3.3 Self-Refrine Augmented RS

To optimize the prompt pipeline, Madaan et al. (2023) proposed a unified framework “Self-Refrine”. As illustrated in Figure 5.3, this framework allows the LLM to generate feedback (step ②) for inferences from the initial generation (step ①) or from the refined generation (step ③). Then, the feedback will be used to construct the prompt, enhancing the performance of subsequent refined generation (step ③). The self-refine loop (step ② and ③) continually iterates until a predefined stopping criterion is satisfied, at which point the LLM halts the process and delivers the inference result from the final prompt as the optimal output. Here, we will not provide a more detailed thorough background on the Self-Refrine framework and refer readers to the origin paper (Madaan et al., 2023).

Let R_{ge} , R_{fb} , and R_{re} be the generation, feedback, and refinement prompt formats for the reduction phase, respectively. Likewise, let S_{ge} , S_{fb} , and S_{re} be those counterparts for the synthesis phase. We utilize the same stop condition f_{stop} for both phases. Let \mathcal{F}_{SR} indicate the Self-Refrine algorithm and llm be the model used to infer generation at each prompt step. In the first phase, the style-free content C from the source X can be obtained by Eq. (5.2). The final generation Y is inferred in the second phase which is given by Eq. (5.3).

$$C = \mathcal{F}_{SR}(X, llm, R_{ge}, R_{fb}, R_{re}, f_{stop}) \quad (5.2)$$

$$Y = \mathcal{F}_{SR}(X, C, llm, S_{ge}, S_{fb}, S_{re}, f_{stop}) \quad (5.3)$$

5.4 Experiments

5.4.1 Setup

We conducted experiments on two benchmark datasets for SST: Yelp (Xiang et al., 2015) and Amazon (Li et al., 2018) reviews. Every dataset combines 1,000 examples

which are split into two groups, 500 sentences for $neg \rightarrow pos$, and another 500 for $pos \rightarrow neg$. Considering the time and computing cost, We choose the LLaMA2-13B (Touvron et al., 2023) as the backbone during inference. The model is experimented with Pytorch on one NVIDIA A6000 GPU (48GB memory).

The predefined stopping conditions consist of two criteria: 1) the number of iterations reaches the threshold η ; 2) the latest inference satisfies the transfer objective as determined by feedback and keyword matching. The final determination is reached by combining these two criteria using a logical AND operation. For each pipeline enhanced with the Self-Refine algorithm, the threshold η is set as 2. The primary hyperparameters configured for the backbone LLM include a top- p value of 0.9 and a temperature of 0.6.

As all inferences are conducted by leveraging the Self-Refine algorithm, for both baseline and our method, we design the initial generation prompt, feedback prompt, and refine prompt, respectively. In each phase, we design 2-shots for every prompt format in Eqs. (5.2) and (5.3). Appendix C illustrates more detailed prompt formats.

5.4.2 Evaluation Metrics

To provide a comprehensive comparison of the effectiveness of various methods, we performed two types of evaluations: automated evaluation and human evaluation.

(1) Automatic Evaluation

Following the settings of Suzgun et al. (2022), we used three aspects of evaluation metrics. The first is content preservation, which consists of reference-SacreBLEU (r-sB) and self-SacreBLEU (s-sB) scores. Here, r-sB and s-sB measure the distance from the generated sentence to the ground truth reference, and the degree to which the model directly copies the source, respectively. The second is transfer strength, which is scored by using accuracy (Acc) on the target style of the generations. The last is the fluency of generated texts consisting of average token-level perplexity (t-PPL) and average sentence-level perplexity (s-PPL). Furthermore, we add a new metric, the rate of r-sB against s-sB, named r/s-sB for evaluating the intent of the trade-off between generating new text and preserving source content during style transfer.

To calculate the r-sB and s-sB scores, we used the evaluator, which is available from the Hugging Face.¹ The Python toolkit for sentiment analysis, named pysentimiento² (Pérez et al., 2021) is utilized to run a sentiment classifier to calculate the Acc. The gpt2-large³ is selected as the language model to compute the t-PPL and s-PPL.

¹<https://huggingface.co/docs/evaluate/index>

²<https://github.com/pysentimiento/pysentimiento>

³<https://huggingface.co/openai-community/gpt2-large>

(2) Human Evaluation

To mitigate the insufficiency of automatic metrics, we also conducted a small-scale in-house human evaluation of the Yelp dataset by assigning the predictions of 50 samples to two reviewers with background knowledge about the domain of the dataset. The evaluation criterion consists of the content preservation capacity, sentiment transfer length, and fluency, and a score range from 1 to 5 is annotated for each aspect⁴. Finally, we average scores from two reviewers for the same example in the test dataset.

Model	Automatic Evaluation						Human Evaluation		
	Acc \uparrow	r-sB \uparrow	s-sB \uparrow	r/s-sB \uparrow	t-PPL \downarrow	s-PPL \downarrow	Content \uparrow	Style \uparrow	Fluency \uparrow
<i>pos \rightarrow neg</i>									
BL	87.4	23.0	44.0	0.523	64	134	3.87	4.05	4.16
RS	85.8	16.1	28.7	0.562	58	110	3.78	3.90	4.15
BL+RS	93.0	21.8	40.1	0.545	61	126	3.93	4.17	4.18
impv. (%)	+6.4	-5.2	-8.9	+4.2	+4.7	+6.0	+2.6	+3.0	+0.5
<i>neg \rightarrow pos</i>									
BL	63.6	16.7	27.3	0.612	33	78	3.34	3.46	3.65
RS	63.4	12.1	19.0	0.637	31	57	3.40	3.59	3.70
BL+RS	72.4	15.6	24.4	0.640	30	70	3.41	3.59	3.69
impv. (%)	+13.8	-6.5	-10.7	+4.6	+9.1	+10.3	+2.1	+3.8	+1.1

Table 5.1: Comparison with the Self-Refine (baseline, represented with BL) on Yelp dataset. The RS indicates the plug-and-play method, and the BL+RS is the method augmenting the BL with RS, that is, replacing the incorrect output of BL with the generation of RS. The **bold** font marks the best performance of each metric. The “impv.” means the improvements of BL+RS against the baseline.

Model	<i>pos \rightarrow neg</i>					<i>neg \rightarrow pos</i>				
	Acc \uparrow	r-sB \uparrow	s-sB \uparrow	r/s-sB \uparrow	t-PPL \downarrow	Acc \uparrow	r-sB \uparrow	s-sB \uparrow	r/s-sB \uparrow	t-PPL \downarrow
CrossAlignment	72.0	7.3	19.3	0.378	224	74.0	8.3	19.3	0.430	190
GPT-J-6B-4s	81.0	25.3	50.5	0.501	107	52.0	21.7	48.7	0.569	82
BL	87.4	23.0	44.0	0.523	64	63.6	16.7	27.3	0.612	33
BL+RS (ours)	93.0	21.8	40.1	0.545	61	72.4	15.6	24.4	0.640	30

Table 5.2: Comparison with related work on the Yelp dataset. The results of CrossAlignment and GPT-J6B-4s are referred to in the work of [Suzgun et al. \(2022\)](#). The **bold** font shows the best performance for each metric in each group (*pos \rightarrow neg* and *neg \rightarrow pos*). \uparrow : Instead of fine-tuning a Roberta model in the related work, we used a third-party sentiment analysis toolkit to calculate the Acc of generations, which is explained in Section 5.4.2.

⁴All annotations are blind, i.e., the reviewers do not know which method was used to make the predictions.

Model	<i>neg</i> \rightarrow <i>pos</i>					<i>pos</i> \rightarrow <i>neg</i>				
	Acc \uparrow	r-sB \uparrow	s-sB \uparrow	r/s-sB \uparrow	t-PPL \downarrow	Acc \uparrow	r-sB \uparrow	s-sB \uparrow	r/s-sB \uparrow	t-PPL \downarrow
BL	30.4	32.5	64.4	0.505	63	34.0	40.8	76.5	0.533	71
RS	32.6	30.6	58.6	0.526	60	37.8	31.4	57.4	0.547	51
BL+RS	38.2	31.1	60.7	0.513	58	45.4	38.7	70.1	0.552	62
impv. (%)	+25.7	-4.3	-5.7	+2.0	+7.9	+33.5	-5.1	-8.4	+5.5	+12.7

Table 5.3: Comparison with the Self-Refine (baseline, represented with BL) on Amazon dataset. The RS indicates the plug-and-play method, and the BL+RS is the method augmenting the BL with RS, that is, replacing the incorrect output of BL with the generation of RS. The **bold** font shows the best performance for each metric. The “impv.” means the improvements of BL+RS against the baseline.

5.4.3 Results

Table 5.1 shows the performance comparison with the Self-Refine baseline on the Yelp dataset. Except for the r-sB, and s-sB scores, our method (BL+RS) which is enhanced by plug-and-play can improve the performance over the baseline by both automatic and human evaluations. As Suzgun et al. (2022) mentioned, the *neg* \rightarrow *pos* transfer is more challenging than that of *pos* \rightarrow *neg* in all metrics, except for the perplexities, obtained for *pos* \rightarrow *neg* far exceeds that for *neg* \rightarrow *pos*. except for r/s-B, t(s)-PPL. The improvements obtained by our plug-and-play method for *neg* \rightarrow *pos* (by Acc, r/s-B, s-PPL, Style, and Fluency) are larger than those of the counterparts for *pos* \rightarrow *neg*.

We can see from Table 5.1 that our RS can improve the content score in human evaluation for both transfer directions, while BL+RS is worse than the baseline (BL) for the r-sB and s-sB in automatic metrics. One possible reason is that the LLM generates more creative content by two phrases prompting in the RS method. Another factor is that the two objectives, transferring sentiment style and preserving content are trade-offs and often conflict. The inherent flaws of automatic metrics result in the inconsistency with human evaluation, as discussed by Mir et al. (2019), the BLEU only measures n-gram overlaps and does not take the style transfer into account is accompanied by changes of words. It is worth noting that our RS obtains a worse entire performance than BL. This demonstrates that RS is only suitable for transferring challenging cases.

In Table 5.2, we also compare the performance of baseline and our method on the Yelp dataset with several related works including one supervised learning-based method, CrossAlignment (Shen et al., 2017), and one prompt-based methods (Suzgun et al., 2022). Consistently, our method (BL+RS) performs better on most metrics.

We also conducted the experiments by using the Amazon dataset. Table 5.3 shows the results of our RS plug-and-play method compared with the baseline. Consistent with the results on Yelp presented in Table 5.1, the RS improves the baseline by enhancing accuracy and increasing the probability of the model generating sentences

Style	Reduction (%)	Synthesis (%)	Self-refine (%)
<i>neg</i> \rightarrow <i>pos</i>			
$s_o = neg$	230 (72.8)	63 (21.7)	54 (17.1)
$s_i = neg \quad s_o = neu$	68 (21.5)	42 (14.5)	37 (11.7)
$s_o = pos$	18 (5.7)	185 (63.8)	225 (71.2)
$s_i = neg$	316	290	316
$s_o = neg$	45 (31.3)	9 (5.6)	9 (6.2)
$s_i = neu \quad s_o = neu$	82 (56.9)	46 (28.4)	73 (50.7)
$s_o = pos$	17 (11.8)	107 (66.0)	62 (43.1)
$s_i = neu$	144	162	144
$s_o = neg$	15 (37.5)	0 (0)	0 (0)
$s_i = pos \quad s_o = neu$	12 (30.0)	1 (2.1)	3 (7.5)
$s_o = pos$	13 (32.5)	47 (97.9)	37 (92.5)
$s_i = pos$	40	48	40
<i>pos</i> \rightarrow <i>neg</i>			
$s_o = neg$	1 (16.7)	35 (83.3)	5 (83.3)
$s_i = neg \quad s_o = neu$	3 (50.0)	3 (7.2)	1 (16.7)
$s_o = pos$	2 (33.3)	4 (9.5)	0 (0)
$s_i = neg$	6	42	6
$s_o = neg$	6 (16.2)	129 (66.5)	16 (43.2)
$s_i = neu \quad s_o = neu$	26 (70.3)	46 (23.7)	20 (54.1)
$s_o = pos$	5 (13.5)	19 (9.8)	1 (2.7)
$s_i = neu$	37	194	37
$s_o = neg$	35 (7.7)	211 (79.9)	378 (82.7)
$s_i = pos \quad s_o = neu$	165 (36.1)	10 (3.8)	14 (3.1)
$s_o = pos$	257 (60.2)	43 (16.3)	65 (14.2)
$s_i = pos$	457	264	457

Table 5.4: Distribution of the style of input and output pairs during every transfer phase on Yelp data. Self-refine is the baseline that directly transfers the input to the target. The background indicates the number and rate of correct results in each transfer phrase. The **bold** in each column refers to the marginal distribution of the input.

with lower perplexity. Similarly, in terms of BLEU scores, RS demonstrates the ability to generate more diverse sentences. We attribute this to the two-step LLM prompting process, which effectively amplifies the randomness of the model’s generation.

To evaluate the effectiveness of the RS pipeline, we conducted sentiment classification on the inputs and outputs of each reasoning step. This analysis enabled us to examine whether the model followed the intended step-by-step process in generating the target sentences. Table 5.4 shows the number of style texts in each of the three transfer phrases, Reduction, Synthesis, and Self-Refine for *neg* \rightarrow *pos*, and *pos* \rightarrow *neg* in Yelp data set. In Table 5.4, s_i and s_o indicate the input and output style, respectively, in each

Model	Pipeline	Acc \uparrow	r-sB \uparrow	s-sB \uparrow	r/s-sB \uparrow	t-PPL
<i>neg \rightarrow pos</i>						
Mistral-7B	BL	82.0	14.1	15.9	0.883	28
	RS	74.8	11.9	15.0	0.789	30
	BL+RS	86.4	13.7	15.3	0.897	27
impv. (%)		+5.4	-2.8	-3.8	+1.6	+3.6
Gemma-7B	BL	52.4	21.2	44.3	0.479	38
	RS	33.6	17.1	38.9	0.438	28
	BL+RS	58.6	20.5	42.3	0.485	38
impv. (%)		+11.8	-3.3	-4.5	+1.3	+7.9
LLaMA-7B	BL	75.4	7.9	8.8	0.901	23
	RS	76.0	7.8	8.8	0.888	26
	BL+RS	81.8	7.7	8.2	0.938	22
impv. (%)		+8.5	-2.5	-6.8	+4.1	+4.3
<i>pos \rightarrow neg</i>						
Mistral-7B	BL	95.6	14.2	19.9	0.715	46
	RS	93.8	11.0	14.8	0.742	58
	BL+RS	97.0	14.2	19.4	0.730	45
impv. (%)		+1.5	+0	-2.5	+2.1	+2.2
Gemma-7B	BL	88.8	22.6	44.2	0.513	61
	RS	81.0	20.6	38.4	0.536	56
	BL+RS	92.6	22.3	41.8	0.534	59
impv. (%)		+4.3	-1.3	-5.4	+4.1	+3.3
LLaMA-7B	BL	97.0	7.9	9.8	0.806	32
	RS	97.6	7.1	9.3	0.763	41
	BL+RS	98.6	7.8	9.7	0.809	32
impv. (%)		+1.6	-1.3	-1.0	+0.4	0

Table 5.5: Comparison with the Self-Refine (baseline, represented with BL) on Yelp dataset by different LLMs. The RS indicates the plug-and-play method, and the BL+RS is the method augmenting the BL with RS, i.e., replacing the incorrect output of BL with the generation of RS. The **bold** font refers to the best performance of each metric. The “impv.” means the improvements of BL+RS against the baseline.

phrase.

As shown in Table 5.4, the number of inputs classified into neutral in *neg \rightarrow pos* case (144) is larger than those of *pos \rightarrow neg* (37). This shows that *neg \rightarrow pos* case includes more ambiguous inputs than *pos \rightarrow neg*, resulting in poor performance. We can also see from Table 5.4 that the synthesis phrase successfully transfers 66.0% neutral texts to the positive style in the *neg \rightarrow pos* task, and 66.5% neutral texts to the negative style in the *pos \rightarrow neg* task in Table 5.4, while the baseline (Self-Refine) of these are 43.1% and, 43.2%, respectively. This indicates the effectiveness of our RS

pipeline.

To explore the performance of the baseline and our RS pipeline across different LLM environments, we conducted experiments on the Yelp dataset using three distinct LLMs for each prompting variant. Table 5.5 illustrates the performance with different LLMs for both transfer directions ($neg \rightarrow pos$, and $pos \rightarrow neg$) on Yelp dataset. We explored the experiments with three popular open-source LLMs (Mixtral, Gemma, and LLaMA with the same 7B size). For a fair comparison, we use the Ollama⁵, a tool for running LLMs in local, to infer all results. As shown in Table 5.5, the overall performance obtained by the baseline is the worst among the three models. In contrast, our BL+RS shows the improvement except for **r-sB** and **s-sB** in both $neg \rightarrow pos$ and $pos \rightarrow neg$.

$neg \rightarrow pos$	
Case 1	you won't find a better selection in scottsdale.
Human	you won't find a better selection in arizona.
BL (✗)	You won't find a better selection in scottsdale.
RS (✓)	You will find the best selection of things in scottsdale.
Case 2	then he proceeded to tell me how to fix it.
Human	He told me kindly how to fix it.
BL (✗)	He told me how to fix it.
RS (✗)	Then he told me how to fix the problem, and I did.
$pos \rightarrow neg$	
Case 3	a great stop if you want nice things at a steal.
Human	a great stop if you want bad things at scammy prices
BL (✗)	a great stop if you want nice things at a steal.
RS (✓)	a terrible place to shop if you want to buy things at a good price.
Case 4	if you're young and are into sports this is the place for you.
Human	no matter your age, you won't like it
BL (✗)	if you're old and are into sports this is the place for you.
RS (✗)	This is a place for young people who are into sports.

Table 5.6: The examples obtained by our RS pipeline and baseline (BL) for challenging SST cases in the $neg \rightarrow pos$ and $pos \rightarrow neg$ tasks. The ✓ and ✗ indicate the text is transferred correctly and incorrectly, respectively.

5.4.4 Case Study

Table 5.6 shows the results obtained by our reduction-synthesis (RS) method and baseline (BL) in four challenging SST cases. The examples shown in Table 5.6 are ran-

⁵<https://github.com/ollama/ollama>

domly selected from the challenging cases on the Yelp dataset. As illustrated in Case 1, with the input “you won’t find a better selection in scottsdale.”, the baseline direct prompting struggles to capture and express the deeper intent conveyed by the LLM (the speaker’s dissatisfaction with “Scottsdale”). Consequently, the baseline merely replicates the original input. In contrast, the RS pipeline generates a sentence incorporating the phrase “best selection,” effectively transforming the negative sentiment into a positive one.

As shown by Case3 in Table 5.6, the human-annotated reference employs irony to transform the positive aspects of the input sentence into a negative expression. Although the RS pipeline does not utilize such advanced rhetorical techniques, it effectively replaces “great” with “terrible” and omits “nice” before “things”, resulting in a more fluent final sentence. This aligns with the findings from the automatic evaluation, where RS consistently achieves lower perplexities.

5.5 Summary

In this work, we proposed a simple, yet effective plug-and-play method, Reduction-Synthesis, to augment the base LLM for the SST task, especially for the challenging transfer cases. The experimental results on two datasets demonstrate that our transfer method is effective for challenging SST cases where the baseline methods perform poorly. Our code is available online⁶.

5.5.1 Contribution

For the SST task, in contrast to previous prompting methods which directly give the examples or instructions to transfer input to target generation, our approach separates the straightforward Seq2Seq generation into two innovative phases: (1) **Reduction** phase which generates a style-free sequence for a given text, and (2) **Synthesis** phase which generates the target text by leveraging the sequence output from the first phase.

By analyzing the language model of SST, we regard the combination of these two steps as a lower bound for the cultivated target of optimization. This allows us to explain, from a probabilistic perspective, how our proposed Plug-and-Play method can be applied to more challenging transfer cases.

5.5.2 Limitation

The performance obtained by our approach is subject to the quality of the middle style-free sequence during the two-step prompt inference. Moreover, carefully crafted

⁶<https://github.com/codesedoc/RS4SST>

prompt formats are necessary for outstanding generation.

In comparison to earlier prompting methods, our Plug-and-Play approach is more akin to a Chain-of-Thought strategy. However, achieving the final target result through a series of incremental corrections necessitates multiple inferences, thereby significantly increasing the overall processing time. Additionally, while the self-refine algorithm enhances performance, it also introduces a corresponding increase in time complexity.

5.5.3 Future Work

Considering the limitations mentioned above, there are some valuable future works including (i) investigating effective generation methods in both two phases, (ii) applying our approach to transfer other text styles, and (iii) exploring more robust prompt templates or pipelines.

As one of the promising future works, the next chapter will explore the performance of different LLMs on the SST task using various prompt patterns to enhance the quality of prompt templates. Additionally, in the next chapter, we further optimize the feedback step of the self-refine process, thereby enhancing the overall stability of the pipeline’s performance.

Chapter 6

Static and Dynamic Prompting for SST

We recall that we reviewed the application of LLMs and prompting methods to TST-related tasks in Chapter 2. As illustrated in Figure 2.7, we introduced three types of prompting, which can be classified into two paradigms: static prompting and dynamic prompting. In Chapter 5, we introduced a Plug-and-Play strategy aimed at improving the baseline performance in handling complex transfer cases during inference. In this chapter, we will undertake a more comprehensive investigation into the performance of two prompting paradigms, various prompt templates, and different LLMs for SST.

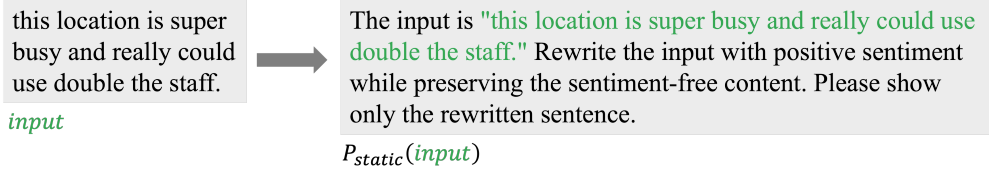
6.1 Motivation

With a large scale size of datasets and the number of parameters, pre-trained models have obtained outstanding improvement on many tasks with few or even zero-labeled data, especially the performance of LLM attracting many interests of researchers. Furthermore, various types of LLMs, along with different versions of these models, are undergoing rapid development and frequent iterations.

Based on the assumption, i.e., an arbitrary NLP task can be regarded as a prompt described with Natural Language, the pre-trained language models can be applied to any task by filling input into a prompt template with amazing generalization (Reif et al., 2022; Suzgun et al., 2022; Yang et al., 2023b; Luo et al., 2023; Liu et al., 2024; Mukherjee et al., 2024c). We note that the carefully designed prompt templates often dominate the entire performance. This indicates that it remains under-explored to bridge the gap between complex linguistic phenomena and finite static prompts, especially when metaphor or irony implicitly expresses the sentiment feature.

Current research extensively explores the architecture of LLMs and their reasoning capabilities for SST. However, a comprehensive study comparing the performance of

♣ Static Prompt



♣ Dynamic Prompt

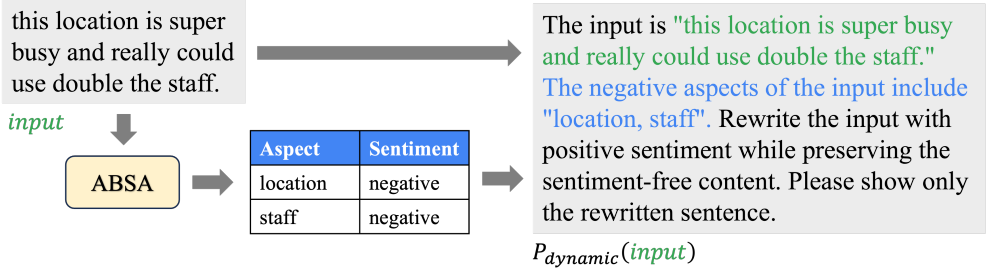


Figure 6.1: An input example of our static and dynamic prompts

static prompting and dynamic prompting under varying conditions remains lacking. Furthermore, previous studies have primarily focused on static prompting or pipelines composed of multiple static prompting steps. Previous research on dynamic prompting is still limited.

In this work, we focus on a dynamic prompt instead of a static prompt for sentiment style transfer (SST) to assist the backbone LLM which learns more input-related context. We utilize an aspect-based sentiment analysis (ABSA) model for a dynamic prompt. To alleviate the common hallucination problem, we also explore two variants of dynamic prompts by leveraging the Self-Refine framework (Madaan et al., 2023) and use the same ABSA model to construct dynamic feedback of each generation from the LLM, which differs from Chapter 5, where feedback for LLMs was generated using static examples within prompt templates.

6.2 Methodology

6.2.1 Dynamic Prompt Template

In this work, we design a template, P_{static} , for a static prompt, depicted in Figure 6.1. Compared with the static prompt which straightforwardly fills the input into the template, our dynamic prompt template ($P_{dynamic} = f_{prompt}(ABSA(input))$) consists of the origin input, the input-specific aspects, and their sentiment polarities which are predicted by an ABSA model.

As shown in Figure 6.1, given the input “this location is super busy and really could use double the staff.” The dynamic segment is constructed based on the prediction from

Algorithm 1 Dynamic prompting by using Self-Refine framework

Require: Input x , iteration limitation η , Generation Model \mathcal{G}_{llm} , ABSA model \mathcal{M}_{ABSA} , sentiment classifier \mathcal{M}_{senti} , target sentiment s_{target} , dynamic refine prompt constructor $\mathcal{C}_{drp}(\cdot)$.

```
1:  $stop(text, t) = \mathcal{M}_{senti}(text)$  is  $s_{target}$  or  $t > \eta$ 
2:  $prompt_{all} = P_{static}(x)$ 
3:  $y = \mathcal{G}_{llm}(prompt_{all})$ 
4:  $i = 1$ 
5: while true do
6:   if  $stop(y, i)$  then
7:     break
8:   end if
9:    $P_{refine} = \mathcal{C}_{drp}(\mathcal{M}_{ABSA}(y))$ 
10:   $prompt_{all} = prompt_{all} \parallel P_{refine}(y)$ 
11:   $y = \mathcal{G}_{llm}(prompt_{all})$ 
12:   $i = i + 1$ 
13: end while
14: return  $y$ 
```

the ABSA model, as: “The negative aspects of the input include “location, staff”. ”

6.2.2 Self-Refine Augmented Dynamic Prompting

The dynamic prompt can also be constructed as feedback by leveraging the Self-Refine framework where Self-Refine is a specific resolution to mitigate the hallucination issues and is often used in LLMs-based systems (Madaan et al., 2023).

Algorithm 1 shows the implementation of the dynamic refine prompt and the stop condition, i.e., $stop(text, t)$ in the Self-Refine framework. The stop condition is triggered by any of two cases: (1) the sentiment style of the last generation is predicted as the target by the sentiment style classifier, or (2) the iteration counts over the limitation, η , where η is a hyper-parameter.

In contrast to Chapter 5, where we also leveraged the Self-Refine algorithm to assist each prompting step of our reduction-synthesis pipeline, we did not design a specific prompt template for generating feedback from the LLM. To fully utilize the capabilities of the ABSA model, we employed it to reclassify the outputs generated by the LLM.

Based on the classification results, we assess whether the most recent generation meets the transfer requirements, and use the assessment as feedback. The feedback is then employed to construct the refinement prompt in the loop of the Algorithm 1.

It is important to note that using ABSA to predict feedback within the self-refine loop and constructing the refinement prompt can be regarded as a variant of Dynamic

Prompting methods. Therefore, in this chapter, we denote this variant as “Self-Refine”.

In the following section, we focus on comparing the experimental results generated by **Static Prompt (SP)** with those produced by the other three dynamic prompt modes, i.e., **Dynamic Prompt (DP)**, **Self-Refine (SR)**, and **Dynamic Prompt plus Self-Refine (DP&SR)** for the SST task. We also examine the comparative results among various LLMs and their parameter sizes.

6.3 Experiments

6.3.1 Experimental Setup

Similar to Chapter 5, we conducted experiments on two benchmark datasets for the SST task: Yelp (Xiang et al., 2015) and Amazon (Li et al., 2018) reviews. For all four prompt modes, we examine the zero-shot performance on 5 main popular open-source LLMs, which are LLaMA2 (Touvron et al., 2023), LLaMA3 (Meta, 2024), Mistral (Jiang et al., 2023), Gemma (Team, 2024), and Gemma2 (DeepMind, 2024).

All experiments are conducted by setting the zero-shot prompts. Similar to Chapter 5, we use the Ollama tool with consistent hyperparameter settings across all LLMs to infer the results. To meet the requirements mentioned in **Algorithm 1**, we use two libraries, PyABSA¹ (Yang et al., 2023a) and pysentimiento libraries for the ABSA model and sentiment classifier, respectively. The upper bound of iteration, η is set to 2 for SR and DP&SR variants.

6.3.2 Evaluation Metrics

Automatic Evaluation. We used the same five evaluation metrics as we have conducted in Chapter 5.3. The reference SacreBLEU (r-sB) is used to assess the entire transfer performance. The self-SacreBLEU (s-sB) evaluates the strength of content preservation. The style transfer strength is scored by using accuracy (Acc) on the target style of the generations. The average token-level perplexity (t-P) and average sentence-level perplexity (s-P) are used to measure the fluency of the output.

LLM-based Evaluation. It has been demonstrated that prompting LLM to evaluate the performance of generation task instead of the expensive human evaluation (Kocmi and Federmann, 2023; Mukherjee et al., 2024c). Similar to the settings in the previous work conducted by Mukherjee et al. (2024c), we design three prompts to evaluate the sentiment transfer strength, content preservation capacity, and fluency by leveraging the GPT-4 (OpenAI, 2024). The prompt templates used to evaluate these three aspects are illustrated by Figure 10, 11, and 12 in Appendix C. The other prompt templates

¹<https://github.com/yangheng95/PyABSA>

Model	Prompt	<i>neg</i> \rightarrow <i>pos</i>					<i>pos</i> \rightarrow <i>neg</i>				
		Acc \uparrow	r-sB \uparrow	s-sB \uparrow	s-P \downarrow	t-P \downarrow	Acc \uparrow	r-sB \uparrow	s-sB \uparrow	s-P \downarrow	t-P \downarrow
Dataset = Yelp											
Gemma	SP	77.8	5.8	8.5	108	58	90.8	9.1	15.1	114	74
	DP	79.8	5.0	7.6	93	62	90.0	10.2	18.3	139	73
	SR	88.2	5.4	7.7	113	59	97.2	8.9	14.5	113	75
	DP&SR	91.0	4.5	6.6	95	60	97.6	10.2	17.1	142	72
LLaMA2	SP	64.0	9.6	16.2	80	38	83.8	13.2	23.1	119	53
	DP	53.6	9.5	18.1	82	40	78.0	12.8	25.7	115	53
	SR	79.0	9.0	14.3	82	37	96.2	12.9	21.9	119	53
	DP&SR	67.8	8.6	14.4	80	39	93.8	12.7	24.1	121	55
Mistral	SP	48.0	4.6	6.5	77	47	82.2	8.0	12.1	104	54
	DP	53.0	4.9	6.7	101	51	82.4	7.7	11.4	118	59
	SR	76.8	3.9	5.6	84	49	91.8	7.5	11.8	108	56
	DP&SR	80.4	4.2	5.5	91	52	92.0	7.1	10.7	118	60
Dataset = Amazon											
Gemma	SP	78.0	6.7	9.5	107	71	76.4	12.8	16.9	117	78
	DP	79.0	6.0	9.1	102	71	75.4	13.1	17.9	152	76
	SR	88.6	6.3	8.9	114	72	88.0	12.1	15.8	113	77
	DP&SR	90.8	5.6	8.3	110	71	86.6	12.3	16.3	137	74
LLaMA2	SP	61.8	11.7	19.0	83	40	64.6	19.2	26.7	81	43
	DP	49.0	13.2	22.5	84	44	59.2	20.8	30.3	188	47
	SR	77.2	10.0	16.0	80	39	86.8	17.6	23.8	82	44
	DP&SR	64.2	11.3	18.5	90	42	81.0	19.4	27.5	95	47
Mistral	SP	51.2	5.6	7.7	78	48	69.6	10.0	12.1	87	55
	DP	53.0	5.2	7.4	93	55	70.0	8.8	10.8	97	61
	SR	79.6	4.3	6.4	88	53	85.4	9.1	11.2	89	57
	DP&SR	82.6	4.5	6.4	96	56	85.8	8.2	10.0	98	60

Table 6.1: Results obtained by different LLMs on Yelp and Amazon data for *neg* \rightarrow *pos* and *pos* \rightarrow *neg*. All LLMs have the same size of parameters (7B). “SP”, “DP” and “SR” refer to “Static Prompt”, “Dynamic Prompt”. and “Self-Refine” prompt settings, respectively. The scores with bold font highlight the best performances within each model.

proposed by related works are implemented in our source code². Due to the cost limit, we evaluate the results obtained by Gemma2 and LLaMA3 on the first fifty samples from the Yelp dataset for transfer *neg* \rightarrow *pos* in Table 6.2.

²<https://github.com/codesedoc/DP4SST>

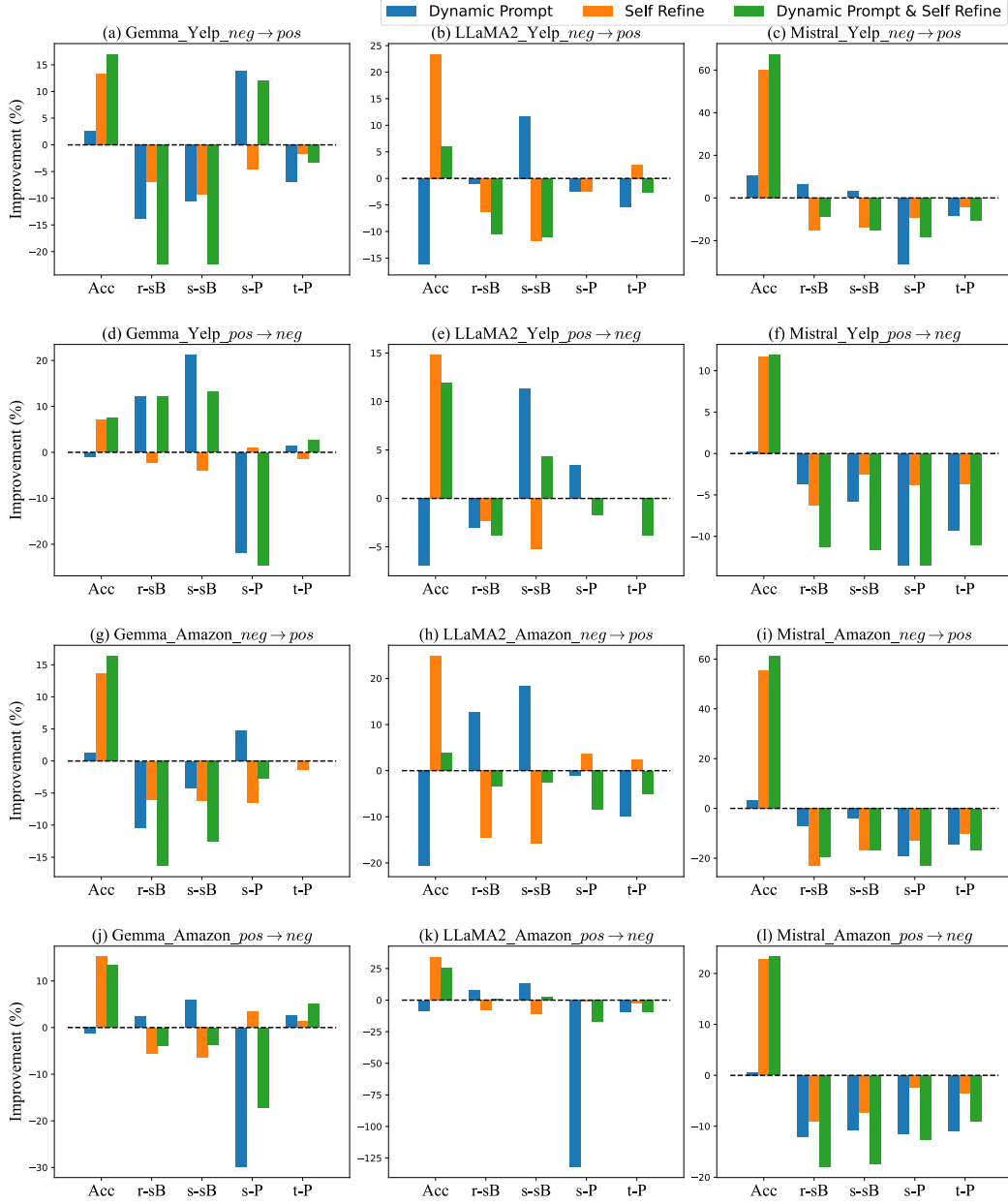


Figure 6.2: Improvement by comparing the three dynamic variances, named “Dynamic Prompt”, “Self-Refine”, and “Dynamic Prompt & Self-Refine” with the baseline “Static Prompt”. The X-axis refers to each metric, and the Y-axis indicates an improvement (%). Each subplot shows the comparison by the same LLM, dataset, and transfer task. All LLMs have the same size of parameters (7B).

6.3.3 Results

Table 6.1 shows the results obtained by three LLMs, i.e., Gemma, LLaMA2, and Mistral with 7B size under four prompt settings. The static prompt, SR, is regarded as our baseline. Figure 6.2 illustrates the improvement of the other three variants referring to the baseline on each metric. We can see from Table 6.1 that except for the r-sB and s-sB, the dynamic prompt modes improve the performance. Figure 6.2 indi-

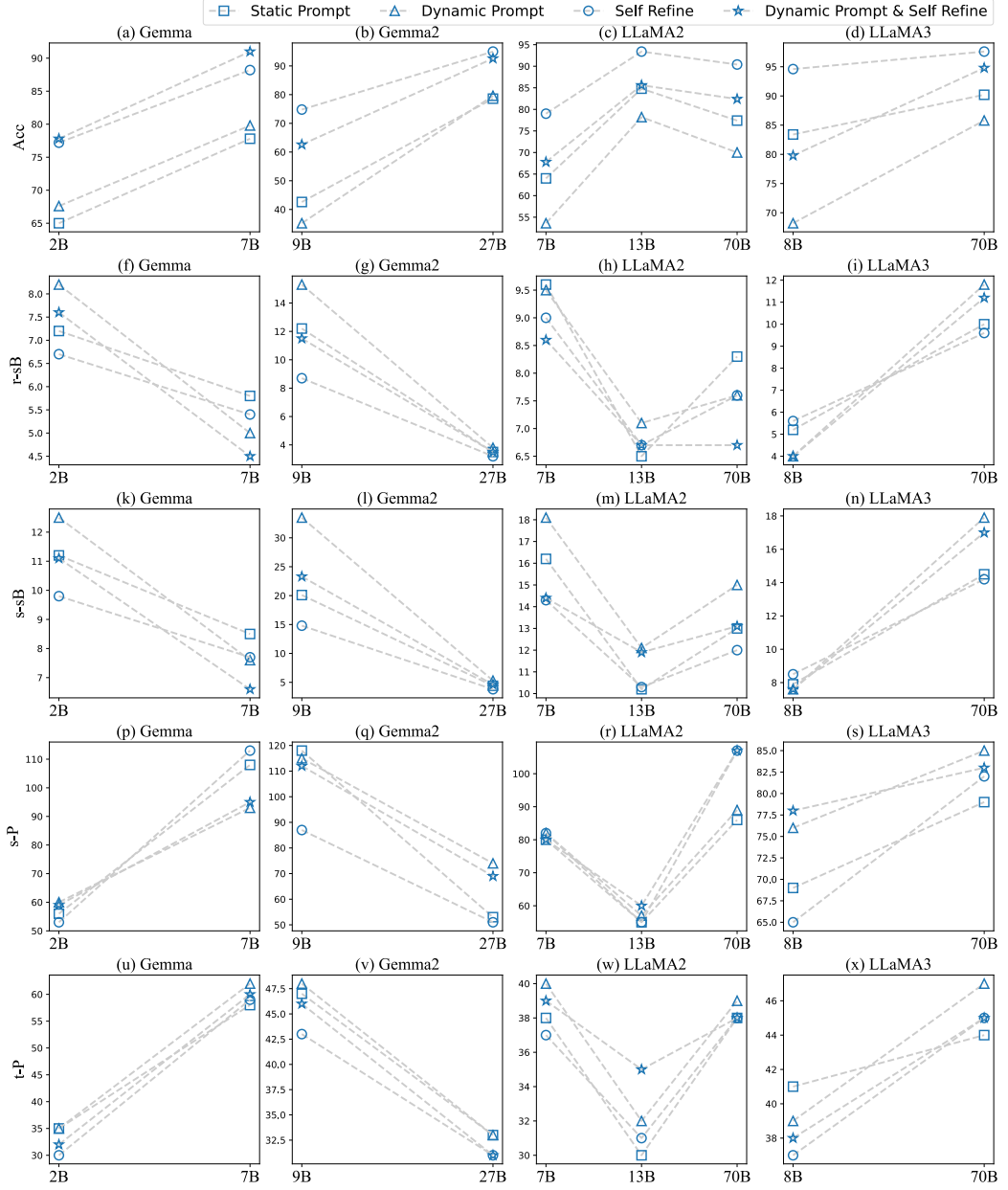


Figure 6.3: SST performance against the different parameter sizes of LLM. The X-axis shows the parameter size of each LLM and the Y-axis indicates metrics, Acc, r-sB, s-sB, and s-P. Four LLM, Gemma, Gemma2, LLaMA2, and LLaMA3, are applied to transfer *neg* \rightarrow *pos* on the Yelp dataset.

cates that good performance in ACC often comes at the expense of the BLEU score. Moreover, compared to the SP, other dynamic modes improve multiple metrics using Gemma or LLaMA2, while only providing greater ACC with Mistral. The consistent improvement in Acc demonstrates that the dynamic prompt constructed by the ABSA model for each input can help the model transfer correctly the input to the target sentiment style. The observation that the increase of ACC decreases BLEU or perplexity score indicates more robust automatic evaluation metrics are necessary to examine the trade-off between style transfer and content preservation.



Figure 6.4: SST performance against the prompt protocols which are used in each static and dynamic prompt template. Three LLMs, i.e., Gemma, Gemma2, and Mistral are used to transfer $neg \rightarrow pos$ on the Yelp dataset. The X-axis indicates each method and the Y-axis shows Acc, r-sB, s-sB, s-P, and t-P improvements.

Figure 6.3 illustrates the performance among different parameter sizes of LLM for transfer $neg \rightarrow pos$ on the Yelp dataset. We conducted experiments using four different LLMs i.e., Gemma, Gemma2, LLaMA2, and LLaMA3. We can see from Figure 6.3 that comparing the sizes for LLM, except for the LLaMA2-13B, the larger model will improve the performance of Acc. As Acc increases, a larger Gemma size results in worse performance on other metrics. In contrast, a larger Gemma2 size can achieve better perplexity performance, while resulting in lower BLEU scores. The larger size of LLaMA3 obtains the higher BLEU score, while the worse score of perplexity.

The result of the LLM-based evaluation by using GPT-4 is shown in Table 6.2. Overall, the DP is worse than the SP in all LLMs. One possible reason for this is that the DP is more complex than the straightforward static prompt, which results in an unclear definition of the transfer task for Gemma2, especially by setting a zero-short prompt. The self-refine algorithm can help the two dynamic variants (SR or DP&SP) outper-

Model	Prompt	Content	Transfer	Fluency
Gemma2-9B	SP	69.8	49.3	66.6
	DP	63.4	39.8	60.5
	SR	72.2	61.3	75.52
	DP&SR	65.7	53.3	69.96
Gemma2-27B	SP	66.36	67.7	75.0
	DP	65.6	67.2	73.5
	SR	69.6	68.6	77.8
	DP&SR	68.5	71.1	78.06
LLaMA3-8B	SP	77.59	75.35	81.77
	DP	73.1	62.9	70.2
	SR	63.8	67.7	70.38
	DP&SR	68.4	66.0	70.26
LLaMA3-70B	SP	86.25	85.2	88.25
	DP	80.5	80.75	85.75
	SR	80.25	80.5	84.0
	DP&SR	78.5	77.25	80.5

Table 6.2: Automatic evaluation by using GPT-4 with different prompt types on three aspects including “Content” (performance for preserving sentiment-independent content), “Transfer” (strength of transferring to target sentiment), and “Fluency” (fluency of generated text). All results are grouped by the model with different parameter sizes. The scores with bold font highlight the best performances within each model.

form the SP by using Gemma2, while in LLaMA3, SP attains better results than all dynamic models. It is worth exploring why LLaMA3 can not improve performance like Gemma2, even with the help of the Self-Refrine framework in future work.

Note that when using LLM, different prompt templates can lead to large differences in results. We thus examine the performance of the prompt templates shown in Figure 6.1, and other five variants from the related works which are “Augment” (R.A.) proposed by Reif et al (Reif et al., 2022), and those, proposed by Suzgun et al. (2022), including “Vanilla” (S.V.), “Contrastive” (S.C.), “Negation_v1” (S.N.1), and “Negation_v2” (S.N.2). Figure 6.4 depicts the performance of the six prompt templates by using three LLMs, Gemma, LLaMA2, and Mistral with the same size of 7B for the transfer $neg \rightarrow pos$ on the Yelp dataset. All results are shown in improvement rates of the scores obtained by three dynamic prompt modes against the baseline, SP.

We found that all templates except for the DP pipeline, increase Acc, especially, the S.N.1 prompt style using DP&SR pipeline has significant improvement, i.e., more than 300%. However, for other metrics, the improvement is not significant, even the dynamic variants perform worse than the baseline by using templates proposed by Suzgun et al. (2022), i.e., S.V., S.C., S.N.1, and S.N.2.

6.4 Summary

In this chapter, we examined the performance of three dynamic prompt modes against the baseline, static prompt mode. We conducted massive experiments to analyze the effectiveness of LLMs, against the parameter sizes of each LLM and prompt templates.

6.4.1 Contribution

Throughout the experiments, We examine the performance of the type and scale of backbone models by applying a wide range of LLMs with different parameter sizes. The main contributions of this paper can be summarized as follows:

- (1) **Prompting Paradigm.** Experimental results obtained by LLMs demonstrate that the dynamic prompt constructed with the sentiment aspects improves the transfer strength for the SST task
- (2) **Evaluation Metric.** Although LLM-based auto-evaluation shows significant coherence with human evaluation, we found that more robust automatic evaluation metrics are needed to assess the trade-off between style transfer and content preservation.
- (3) **Scaling Law.** Similar to lots of previous attempts ([Kaplan et al., 2020](#)), the larger model improves consistently in all LLMs for transferring sentiment regardless of static or dynamic prompts.

6.4.2 Limitation

Compared to Chapter 5, although our pipeline does not decompose general transfer into multiple substeps, the self-refine enhanced prompting remains more time-consuming than both SP and DP.

In this study, the construction function of our dynamic prompting (DP) template is designed to consider only the sentiment attributes of the input sentence, which limits its direct applicability to other TST tasks. To this end, the task-specific DP construction function must be redesigned.

6.4.3 Future Work

From an intuitive perspective, it is worth further exploring variations of prompt modes for different text style transfers. To overcome the limitations of DP in content preservation and generation fluency, the construction of the DP template must take into account

multiple factors, rather than being confined solely to style features. Note that, this will inevitably increase the complexity of the prompting process.

Chapter 7

Entangle or Disentangle, which is Better for TST

7.1 Motivation

In the preceding four chapters, we introduced four research studies, each focused on sentiment-related tasks, including PTR and SST. As analyzed in these chapters, fine-tuning-based methods are constrained in both performance and generalizability due to their reliance on parallel datasets. While LLMs-based prompting methods offer greater versatility by eliminating the need for parallel data, prompt engineering requires researchers to possess domain-specific expertise to design effective prompt templates for specific downstream tasks.

In Chapters 5 and 6, we explored various prompting approaches tailored to SST tasks, such as reduction-synthesis plugins and ABSA model-based dynamic prompting. However, compared to simpler and more intuitive baselines, these methods lack the flexibility to be directly extended to other TST sub-tasks, such as FST and AST.

To explore a novel LLM-based approach for arbitrary TST tasks, we revisited the fundamental characteristics of these tasks from two overarching strategies: disentanglement and entanglement. As illustrated in (a) of Figure 7.1a, the disentanglement strategy involves decomposing the input sentence into separate content and style components. The output is subsequently generated by retaining the content while transferring the source style to the target style. In contrast, as depicted in (b) of Figure 7.1b, the entanglement strategy generates the output by integrating the target style with the information extracted from the input without altering the inherent content.

Leveraging strategies based on disentanglement or entanglement is a basic and intuitive solution to the TST tasks. These strategies have been extensively examined in prior research, including training-from-scratch methods (Bao et al., 2019; Dai et al., 2019), fine-tuning-based approaches, as discussed in Chapters 3 and 4, and LLM-based

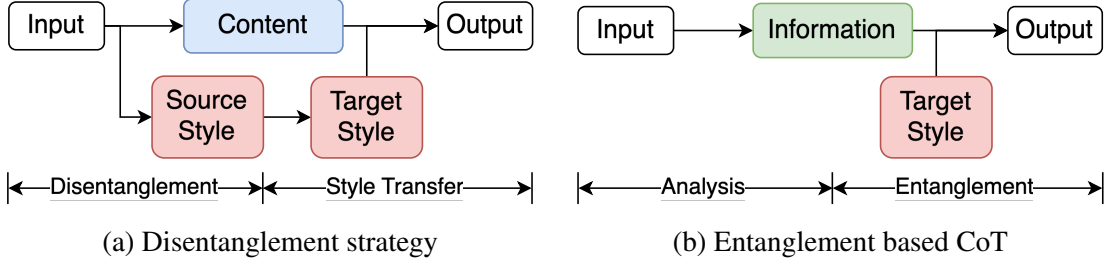


Figure 7.1: Two overarching strategies for TST

prompting techniques (Han et al., 2024). For instance, the seq2seq model employing the disentanglement strategy is trained to learn disentangled representations in the latent space. Similarly, under the entanglement strategy, the decoder integrates controllable style features with the representations of the source sentences to generate the target sentence.

Although previous studies have demonstrated the effectiveness of their disentanglement or entanglement strategies through experimental results, a systematic investigation into which strategy is more effective is still an open problem, particularly when employing LLMs.

In this chapter, to address the aforementioned limitations, we propose two LLM-based pipelines using the disentanglement and entanglement strategies, respectively. To comprehensively compare the performance and generalizability of each pipeline, we conduct experiments on four TST subtasks including SST, FST, AST, and Grammar Style Transfer (GST).

7.2 Method

7.2.1 Constructing Pipelines by Leveraging CoT

To enhance the controllability and logical coherence of LLMs reasoning processes, we design our two pipelines grounded in the Chain-of-Thought (CoT) prompting (Wei et al., 2022), designed to ensure robust performance across a wide range of TST subtasks. Considering the prompting template can be directly constructed by Natural Language to define the expected transfer, each CoT pipeline consists of two steps of prompting which are designed by following the disentanglement and entanglement strategies shown in Figure 7.1a and 7.1b, in (a) and (b) in Figure 7, respectively. Let “[input]” indicate the input sentence with an original style “[ss]”. The target style is represented by “[ts]”. The style to be transferred is referred to as “[style]”. For the two steps of the disentanglement pipeline, we design the prompt templates as follows:

Disentanglement Prompt: Here is a sentence “[input]”. Please analyze which

part expresses $[ss]$, and which is $[style]$ -independent content.

Style Transfer Prompt: Based on the analysis, please revise the sentence to transfer $[ss]$ content to $[ts]$ while preserving the $[style]$ -independent content.

Similarly, the prompt templates for the entanglement pipeline are presented as follows:

Analysis Prompt: Here is a sentence “[*input*]”. Please analyze the information conveyed in this sentence.

Entanglement Prompt: Based on the analysis, please revise the sentence to express a more $[ts]$.

The Chain-of-Thought (CoT) pipelines, designed for interacting with LLMs to address each specific task, are detailed in our code¹.

7.2.2 Assemble Disentanglement and Entanglement

Considering the diversity of TST cases and the inherent flexibility of natural language, we assume that relying exclusively on either a disentanglement- or entanglement-based CoT pipeline may not be enough to handle all scenarios effectively. As depicted in Figure 7.2, the first input sentence can be easily decomposed into a content component, “*Ever since joes has changed hands it’s just gotten*”, and a style component, “*worse and worse.*”. However, the second input sentence presents challenges in explicitly separating content and style in natural language, as it expresses sentiment implicitly, and requires more advanced reasoning capabilities. In such cases, the entanglement-based pipeline may achieve better results.

Figure 13 and 14 in Appendix D presents the prompts and the corresponding model responses when using LLaMA3.2 with disentanglement CoT to transfer these two examples.

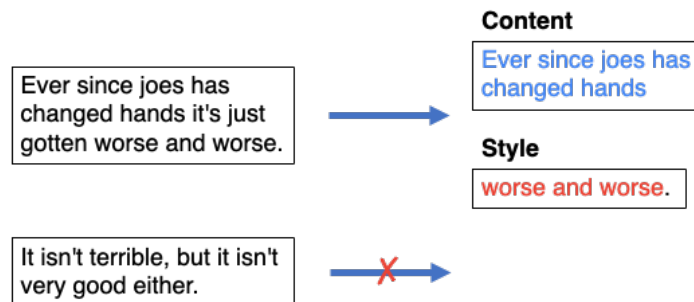


Figure 7.2: Two input sentences processed based on disentanglement.

¹<https://github.com/codesedoc/CoT4TST>

To fully exploit the advantages of both CoT strategies, we adopt the reranking method proposed by [Suzgun et al. \(2022\)](#). For each input, both CoT pipelines are applied, and their outputs are subsequently evaluated with three scores, representing the strength of style transfer, content preservation, and fluency. Different from [Suzgun et al. \(2022\)](#), who calculate these three scores by automatic metrics, we directly use an LLM-based evaluator to evaluate each regularized score by prompting as illustrated in Figure 7.3. All three scores are multiplied to get the total score, the output with the higher total score is regarded as the final generation.

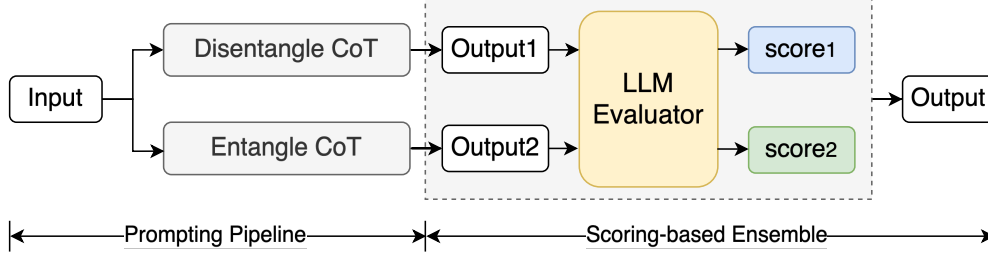


Figure 7.3: An illustration of ensembling disentangle and entangle pipelines by leveraging LLM as scoring evaluator.

7.3 Experiments

7.3.1 Experimental Setup

We conducted experiments on four TST subtasks, which are SST, FST, AST, and GST. The datasets, which have been cleaned by [Suzgun et al. \(2022\)](#), used for these tasks are briefly explained as follows:

- (1) **SST**. Similar to Chapter 5 and Chapter 6, we choose the annotated Yelp and Amazon test datasets for the SST task, where both $neg \rightarrow pos$ and $pos \rightarrow neg$ transfer directions are considered.
- (2) **FST**. Following most of the related works, we use the GYAFC dataset collected by [Rao and Tetreault \(2018\)](#) to evaluate the performance of each variant for the FST task. We focus on the *informal* \rightarrow *formal* transfer.
- (3) **AST**. For the AST task, we leverage a small subset of the dataset proposed by [Xu et al. \(2012\)](#), which is used to translate the plays of Shakespeare to their counterparts written in modern English (*shakespearean* \rightarrow *modern*). For convenience, the subset is named “SHASP”.
- (4) **GST**. The last dataset we selected is JFLEG, which was proposed by [Napoles et al. \(2017\)](#) for the automatic grammatical error correction task. We conducted

Dataset	Transfer	Task	Size
Yelp	$neg \rightarrow pos$	SST	500
Yelp	$pos \rightarrow neg$	SST	500
Amazon	$neg \rightarrow pos$	SST	500
Amazon	$pos \rightarrow neg$	SST	500
GYAFC	$informal \rightarrow formal$	FST	500
JFLEG	$ungrammatical \rightarrow grammatical$	GST	747
SHASP	$shakespearean \rightarrow modern$	AST	599

Table 7.1: The statistics of seven datasets for different TST subtasks

the transfer from ungrammatical sentences to their grammatical counterparts ($ungrammatical \rightarrow grammatical$).

Since Yelp and Amazon each contain two subsets for $neg \rightarrow pos$ and $pos \rightarrow neg$ tasks, respectively, all other datasets involve single-directional transfer. In total, seven TST datasets are used across all experiments. The statistics of these datasets are shown in Tabel 7.1.

As mentioned in Chapter 6, we conducted the result on six popular open-source LLMs, including LLaMA 3, LLaMA 3.1, LLaMA3.2, Gemma, and Gemma 2. To obtain the most accurate scores, we select LLaMA3.3 with 70 billion parameters as the scoring evaluator for the ensemble operation. The scoring prompt templates are listed in Appendix C. Likewise, all models are set up by utilizing the Ollama tool. To focus on investigating the disentanglement and entanglement CoTs, all inferences are conducted in a zero-shot context. During each inference step, the hyperparameters are the same settings as mentioned in Chapter 6.

The automatic evaluation metrics include the five used in Chapter 6: accuracy (Acc), reference-SacreBLEU score (r-sB), self-SacreBLEU score (s-sB), token-level perplexity (t-PPL), and sentence-level perplexity (s-PPL). However, we did not use the general classifier to evaluate accuracy more precisely. Instead, we fine-tuned a standard BERT model with the style labels of sentences in each dataset to serve as a specific classifier for every transfer subtask. Following with the resolution in Chapters 3 and 3, we downloaded the pre-trained parameters of “bert-base-uncased” from Huggingface.

7.3.2 Results

Table 7.2 illustrates the performance of the baseline and two CoT prompt variants (disentangle, and entangle pipelines) across six different LLMs on the Yelp dataset for $neg \rightarrow pos$. The disentangle pipeline consistently outperforms the baseline in terms of Acc, r-sB, and s-sB metrics. Moreover, the entangle pipeline demonstrates superior

Model	Pipeline	Acc \uparrow	r-sB \uparrow	s-sB \uparrow	t-PPL \downarrow	s-PPL \downarrow
Gemma	baseline	80.6	4.67	7.3	37	61
	disentangle	84.6	9.06	15.23	39	74
	entangle	91.4	5.32	7.16	29	42
Gemma2	baseline	74.8	6.46	10.28	46	78
	disentangle	82.8	15.04	28.12	50	100
	entangle	87.8	3.82	6.71	30	50
LLaMA2	baseline	83.2	7.56	12.73	35	64
	disentangle	74.2	13.93	26.45	49	92
	entangle	86.2	8.61	14.67	32	64
LLaMA3	baseline	87.6	7.94	12.48	48	85
	disentangle	85.0	17.62	31.59	54	111
	entangle	90.4	7.47	12.23	35	70
LLaMA3.1	baseline	78.0	9.86	17.26	45	87
	disentangle	78.4	18.7	34.59	57	117
	entangle	80.0	8.88	14.43	33	64
LLaMA3.2	baseline	78.2	7.81	13.64	37	69
	disentangle	76.2	16.48	31.4	47	99
	entangle	76.2	8.81	14.85	32	54

Table 7.2: Results of each pipeline on Yelp (*neg* \rightarrow *pos*) dataset, by using six different LLMs. The **bold** font indicates the best scores in every subgroup.

performance compared to the baseline in Acc, t-ppl, and s-ppl metrics. These results suggest that both the disentangle and entangle pipelines improve style transfer strength. However, the disentangle pipeline outperforms in BLEU scores, whereas the entangle pipeline obtains better perplexities. This indicates that the disentanglement CoT is more effective in producing cleaner, target-like sentences, whereas the entanglement CoT excels in generating sentences with higher fluency.

We assume that for scenarios where content and style can be explicitly disentangled (as shown in Figure 7.2), the disentanglement CoT, where the sentence components are first analyzed and only the style-related elements are modified, facilitates a more logical generation process by the LLM. In contrast, the entanglement CoT, which directly incorporates the target style into the original sentence, resulting the inclusion of content-irrelevant information in the final output.

For cases like the second example in Figure 7.2, where it is challenging to explicitly extract the abstract meaning conveyed by the original sentence, the ensemble approach proves more advantageous. By avoiding the need for explicit representation of abstract content, the ensemble approach minimizes the risk of introducing errors and achieves style transfer more seamlessly and naturally.

Table 7.3 presents the performance of LLaMA3.2 across seven TST datasets. As

Dataset	Pipeline	Acc \uparrow	r-sB \uparrow	s-sB \uparrow	t-P \downarrow	s-P \downarrow	S \uparrow	C \uparrow	F \uparrow
Yelp (<i>neg</i> \rightarrow <i>pos</i>)	baseline	78.2	7.81	13.64	37	69	64.24	59.04	67.73
	disentanglement	76.2	16.48	31.4	47	99	67.84	62.85	70.42
	entanglement	76.2	8.81	14.85	32	54	63.38	57.43	66.19
	ensemble	82.2	11.83	20.62	38	79	73.24	66.81	74.79
Yelp (<i>pos</i> \rightarrow <i>neg</i>)	baseline	81.4	10.56	20.64	50	98	65.88	64.34	67.86
	disentanglement	76.6	18.19	38.25	65	165	60.3	60.16	63.1
	entanglement	84.8	10.89	21.01	46	98	64.28	62.14	66.75
	ensemble	91.4	15.62	29.65	55	116	74.23	72.08	75.08
Amazon (<i>neg</i> \rightarrow <i>pos</i>)	baseline	74.4	11.42	16.37	38	62	61.98	59.56	65.9
	disentanglement	74.6	22.12	34.39	51	105	64.54	62.43	68.66
	entanglement	77.8	9.81	15.14	32	61	65.57	57.91	67.59
	ensemble	81.0	14.11	21.66	37	65	70.37	64.83	72.67
Amazon (<i>pos</i> \rightarrow <i>neg</i>)	baseline	70.6	17.38	24.28	47	87	51.93	55.12	56.31
	disentanglement	62.8	27.26	40.21	61	127	45.52	50.63	50.77
	entanglement	90.4	14.82	21.21	40	76	62.23	56.17	63.37
	ensemble	84.6	19.79	27.81	46	81	65.3	61.52	66.93
GYAFC	baseline	98.8	7.48	4.65	30	50	81.73	76.27	80.5
	disentanglement	92.0	13.31	15.18	35	59	78.03	74.96	77.52
	entanglement	98.8	3.1	2.63	25	39	73.56	60.0	69.31
	ensemble	96.4	7.2	7.6	30	50	81.75	74.3	79.51
JFLEG	baseline	94.24	41.02	34.28	32	47	79.87	79.73	85.57
	disentanglement	87.68	46.38	44.05	40	77	71.77	73.41	77.33
	entanglement	95.18	23.57	19.78	28	46	64.89	62.8	74.83
	ensemble	92.1	41.75	37.74	33	53	77.62	78.16	84.52
SHASP	baseline	97.83	4.95	4.64	39	54	59.81	64.65	65.91
	disentanglement	88.15	11.05	15.45	60	95	61.8	67.42	67.45
	entanglement	98.0	4.32	4.39	34	51	49.15	53.54	57.18
	ensemble	94.82	8.72	10.51	47	72	63.89	69.28	70.43

Table 7.3: Results of each pipeline across seven TST datasets by leveraging LLaMA3.2 as the backbone model. The **bold** font indicates the best scores among each subgroup. t-P and s-P represent t-PPL and s-PPL, respectively. S, C, and F denote the style transfer, content preservation, and fluency scores, as evaluated by LLaMA3.3.

shown in Table 7.3, the ensembled results of the disentanglement and entanglement pipelines achieve a trade-off between BLEU scores and perplexities on all TST sub-tasks consistently. It is noteworthy that for the subsets of Yelp and Amazon, the ensemble variant consistently outperforms the baseline across all metrics.

Similar to Table 7.2, the results presented in Table 7.3 further validate that the disentangle pipeline outperforms the baseline in terms of BLEU scores, while the entanglement variant obtains better Acc and perplexity scores for most of the seven TST datasets. These findings suggest that the disentanglement CoT is particularly adept at decomposing sentence components and generating target sentences, while the entan-

glement CoT is more logically intuitive and excels at generating more natural sentences that align with the target style.

7.4 Summary

In this chapter, we focus on investigating the performance of the CoT prompting pipelines based on disentanglement and ensembling in comparison to the baseline. To assess the generalizability of the prompt variants and the robustness of their performance, we selected seven diverse datasets across four different TST sub-tasks. Furthermore, we utilized six distinct LLMs to establish a range of experimental settings. The code for rebuilding our experiments is available at GitHub².

7.4.1 Contribution

We designed prompt templates specifically for the disentanglement CoT and entangle-ment CoT, respectively. Through extensive experiments, we validated the effectiveness of the prompting pipelines. Notably, our analysis revealed that the disentangle pipeline generates outputs that more closely resemble the reference sentences, while the entangle pipeline excels in terms of fluency.

On the other hand, based on the algorithm proposed by (Suzgun et al., 2022), we propose an ensemble operation to trade-off the performance of the disentangle and entangle pipeline for complex and diverse transfer cases in different datasets. The experimental results demonstrate our ensemble variant can obtain consistently better metrics on different datasets.

7.4.2 Limitation

Although extensive experiments have demonstrated the effectiveness of the disentangle and entangle pipelines, they require additional prompting steps compared to the baseline. Furthermore, the ensemble variant, which involves a larger LLM-based evaluator, necessitates greater computational resources, thereby slowing down the overall inference process.

Secondly, as the methods presented in this chapter focus on the generalization performance of CoT, we did not optimize each pipeline for specific TST subtasks. Consequently, the trade-off ensemble results only surpass the baseline on the Amazon dataset, and in other tasks, they exceed the baseline on only certain metrics.

²<https://github.com/codesedoc/CoT4TST>

7.4.3 Future Work

This chapter has preliminarily demonstrated that CoTs, designed based on two fundamental approaches, can improve the performance of models on TST tasks in various aspects. However, further exploration of fine-tuning algorithms that require minimal parameter updates or the use of RAG strategies to fully exploit the potential of both pipelines represents a valuable direction for future work.

It is another promising avenue for future research to design more effective ensemble methods that combine the strengths of both pipelines to enhance the model’s robustness to complex and diverse transfer cases. Evaluating the results of multiple variants is a fundamental strategy. However, since the ground truth is not accessible in this case, accurately and effectively assessing different outcomes and predicting the optimal result is still a challenging task.

To better evaluate the model’s ability to balance style transfer and content preservation, as well as the generalization capability of different pipelines, it is necessary to explore more effective evaluation methods.

Chapter 8

Conclusion

Due to the diversity of language, TST remains a challenging research area within NLP, as it imposes higher requirements on generated outputs compared to other tasks. For example, unlike paraphrase generation, TST necessitates that the target sentence’s language style be controllable—a feature that is inherently difficult to define formally. Similarly, compared to text-controllable generation, TST requires the target sentence to preserve the content of the original sentence while remaining independent of its style.

As outlined in Chapter 1, the scope of TST sub-tasks varies widely depending on the specific style being addressed. To maintain generalizability, this study focuses on two sentiment-related TST sub-tasks: PTR and SST. Furthermore, in Chapter 7, we extend our methods to a broader range of TST sub-tasks.

Building upon prior research, this study conducts two experiments with PLMs and three with LLMs. For PLMs, we continue to employ a fine-tuning approach for the PTR task. While LLMs are essentially an advanced form of PLMs, distinguished by their larger model sizes and more extensive pretraining datasets, they have achieved transformative advancements in performance and versatility. Consequently, our LLM-focused research emphasizes prompting techniques. To provide a comprehensive analysis, the subsequent sections summarize all aspects of this study from three distinct perspectives.

8.1 Dataset Perspective

For PLM-based TST, datasets not only provide training samples for models but also serve as the foundation for evaluations. The quality and scale of datasets directly impact research progress. To ensure relatively reliable fine-tuning, the studies in Chapters 3 and 4 were conducted on the PPF dataset. However, like other parallel datasets, each input in the PPF dataset has a limited number of references, whereas ideal model-

generated sentences often lack fixed patterns. For LLMs operating under the prompting paradigm, the models exhibit greater flexibility, making the limited references inadequate as robust benchmarks for evaluation and constraining the generalization capabilities of fine-tuned models. Moreover, the lack of datasets that comprehensively capture diverse stylistic features of language restricts TST-related research to a limited scope. Building more general, diverse, and open datasets holds considerable promise for future work.

8.2 Model Perspective

In Chapters 3 and 4, case analyses of model performance were conducted. Although the examples indicate that our model slightly outperforms baseline models, many instances still involve a simple copy of the input sentences. This is likely due to the discrepancy between the training objectives used in pretraining and those in fine-tuning. Enhancing the generalization capabilities of PLMs for diverse TST tasks remains a pressing research challenge.

Similarly, in Chapters 5, 6, and 7, while LLM-based prompting methods no longer rely on parallel datasets, the outputs exhibit significant uncertainty and randomness, placing higher demands on the design of prompt templates. Leveraging Retrieval Augmented Generation (RAG) or Knowledge Graphs to enhance LLMs’ understanding of task-specific characteristics and transformation goals presents a direct and promising avenue for future research.

8.3 Evaluation Perspective

Across all five works discussed in this thesis, two types of evaluations were conducted. Automatic evaluation often fails to fully capture the nuances of generative tasks, while providing quick and objective metric scores. This flaw makes it insufficient for a comprehensive assessment of model quality. In contrast, human evaluation yields more accurate scores but is labor-intensive and inherently subjective, making it challenging to establish consistent standards across different datasets. Even when evaluating the same dataset, different experts may be influenced by the contextual background knowledge associated with individual cases, potentially resulting in inconsistent evaluation outcomes.

A promising alternative is LLM-based automatic evaluation, as introduced in Chapter 6. This approach demonstrates greater robustness and stronger alignment with human evaluation compared to traditional metrics. However, it requires additional prompt engineering and mitigating hallucinations during the evaluation process remains an important area for further research.

Another promising future work is to investigate more effective approaches for evaluating the trade-off between style transfer and content preservation. The most straightforward form of transfer entails directly copying the input as the target output. Although this method may yield a self-BLEU score of 100 and exhibit substantial overlap with the reference content, it is insufficient as a generative model for the TST task. On the other hand, if the generative model introduces excessive randomness when producing sentences with the target style, it may diverge from expectations, either failing to preserve content-independent information during generation or introducing content not present in the input sentence.

The model must achieve a more fine-grained trade-off to enable the controlled generation of precise target sentences. However, the simple combination of BLEU and Accuracy metrics is inadequate for intuitively assessing the model’s performance. As evidenced by the experimental results in Chapter 7, while ensembling enhances the balance between disentangling style transfer capabilities and generating fluent sentences in the entangled pipeline, it lacks a clear and intuitive metric for evaluating the model’s effectiveness in this aspect.

Publications

Peer-reviewed Publications

Sheng Xu, Fumiyo Fukumoto, Jiyi Li, Go Kentaro, and Yoshimi Suzuki. Learning Disentangled Meaning and Style Representations for Positive Text Reframing. In Proceedings of the 16th International Natural Language Generation Conference. pages 424–430. 2023.

Sheng Xu, Yoshimi Suzuki, Jiyi Li, Go Kentaro, and Fumiyo Fukumoto. Decoupling Style from Contents for Positive Text Reframing. In Proceedings of the 30th International Conference on Neural Information Processing. pages 73–84. 2023.

Sheng Xu, Fumiyo Fukumoto, and Yoshimi Suzuki. Reduction-Synthesis: Plug-and-Play for Sentiment Style Transfer. In Proceedings of the 17th International Natural Language Generation Conference. pages 330–343. 2024.

Sheng Xu, Fumiyo Fukumoto, Kentaro Go, and Yoshimi Suzuki. Can Dynamic Prompt Help Sentiment Style Transfer?. In Proceedings of the 31st International Conference on Neural Information Processing. 2024.

Non Peer-reviewed Publications

Sheng Xu, Yoshimi Suzuki, Jiyi Li, Go Kentaro, and Fumiyo Fukumoto. Disentangling Meaning and Style for Positive Text Reframing. 言語処理学会第29回年次大会発表論文集. pages 2820–2824. 2023.

徐勝, 鈴木良弥, 福本文代. Event-Centered Prompting for Text Style Transfer. 言語処理学会第30回年次大会発表論文集. pages 3321–3325. 2024.

徐勝, 鈴木良弥, 福本文代. Disentanglement or Entanglement, which is Better for TST. 言語処理学会第30回年次大会発表論文集. 2025.

Contributions to Works

Bassamtiano Renaufalgi Irnawan, Sheng Xu, Noriko Tomuro, Fumiyo Fukumoto and Yoshimi Suzuki. Claim veracity assessment for explainable fake news detection. In Proceedings of the 31st International Conference on Computational Linguistics, pages 4011–4029, 2025.

Yusei Kato, Sheng Xu, Fumiyo Fukumoto and Yoshimi Suzuki. Correlation Learning with Enhanced Attention Mechanism of BERT for Multi-Label Text Classification. IMCOM 2025.

References

- Bell Allan and Jaworski Adam. 1997. *Language Style as Audience Design*, pages 240–250. Macmillan Education UK, London.
- Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. 2019. [Generating sentences from disentangled syntactic and semantic spaces](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019, Florence, Italy. Association for Computational Linguistics.
- Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. [TweetEval: Unified benchmark and comparative evaluation for tweet classification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.
- Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. [A neural probabilistic language model](#). In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.
- Eleftheria Briakou, Sweta Agrawal, Joel Tetreault, and Marine Carpuat. 2021. [Evaluating the evaluation metrics for style transfer: A case study in multilingual formality transfer](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1321–1336, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Julian Brooke, Tong Wang, and Graeme Hirst. 2010. [Automatic acquisition of lexical formality](#). In *Coling 2010: Posters*, pages 90–98, Beijing, China. Coling 2010 Organizing Committee.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. [Language models are few-shot learners](#). *Preprint*, arXiv:2005.14165.
- Yixin Cao, Ruihao Shui, Liangming Pan, Min-Yen Kan, Zhiyuan Liu, and Tat-Seng Chua. 2020. [Expertise style transfer: A new task towards better communication](#)

- between experts and laymen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1061–1071, Online. Association for Computational Linguistics.
- Keith Carlson, Allen Riddell, and Daniel Rockmore. 2018. [Evaluating prose style transfer with the bible](#). *Royal Society Open Science*, 5(10):171920.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607.
- Zhang Chiyu, Cai Honglong, Li Yuezhong, Wu Yuexin, Hou Le, and Abdul-Mageed Muhammad. 2024. [Distilling text style transfer with self-explanation from llms](#).
- Noam Chomsky. 1957. *Syntactic Structures*. De Gruyter Mouton, Berlin, Boston.
- Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing 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, Florence, Italy. Association for Computational Linguistics.
- Google DeepMind. 2024. [Gemma 2 is now available to researchers and developers](#).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Hao Fu, Yan Wang, Ruihua Song, Tianran Hu, and Jianyun Nie. 2021. [Stylistic retrieval-based dialogue system with unparallel training data](#). *Preprint*, arXiv:2109.05477.
- Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):663–670.
- Hiroyuki Fukasawa, Kentaro Go, Fumiyo Fukumoto, Jiyi Li, and Yuichiro Kinoshita. 2023. Creating a positive reframing dictionary using machine learning. In *HCI International 2023 Posters*, pages 411–417. Springer Nature Switzerland.
- Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. [Image style transfer using convolutional neural networks](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423.

- Kentaro Go, Yuki Moriya, Yuichiro Kinoshita, Jiyi Li, and Fumiyo Fukumoto. 2022. [Happy text entering: Promoting subjective well-being using an input method for presenting positive words and phrases](#). In *Proceedings of the 33rd Australian Conference on Human-Computer Interaction*, page 153–158. Association for Computing Machinery.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. [Generative adversarial nets](#). In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Jingxuan Han, Quan Wang, Zikang Guo, Benfeng Xu, Licheng Zhang, and Zhendong Mao. 2024. [Disentangled learning with synthetic parallel data for text style transfer](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15187–15201, Bangkok, Thailand. Association for Computational Linguistics.
- Jingxuan Han, Quan Wang, Licheng Zhang, Weidong Chen, Yan Song, and Zhendong Mao. 2023. [Text style transfer with contrastive transfer pattern mining](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7914–7927, Toronto, Canada. Association for Computational Linguistics.
- Ruining He and Julian McAuley. 2016. [Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering](#). In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee.
- Francis Heylighen, Jean-Marc Dewaele, and Léo Apostel. 1999. [Formality of language: definition, measurement and behavioral determinants](#).
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. [Denoising diffusion probabilistic models](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.
- Eduard Hovy. 1987. [Generating natural language under pragmatic constraints](#). *Journal of Pragmatics*, 11(6):689–719.
- Yahao Hu, Wei Tao, Yifei Xie, Yi Sun, and Zhisong Pan. 2023. [Token-level disentanglement for unsupervised text style transfer](#). *Neurocomputing*, 560:126823.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. [Toward controlled generation of text](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.
- Fei Huang, Zikai Chen, Chen Henry Wu, Qihan Guo, Xiaoyan Zhu, and Minlie Huang. 2021. [NAST: A non-autoregressive generator with word alignment for unsupervised text style transfer](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1577–1590, Online. Association for Computational Linguistics.

- Xun Huang and Serge Belongie. 2017. [Arbitrary style transfer in real-time with adaptive instance normalization](#). In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519.
- Somayeh Jafaritazehjani, Gwénolé Lecorvé, Damien Lolive, and John Kelleher. 2020. [Style versus content: A distinction without a \(learnable\) difference?](#) In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2169–2180, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- F Jelinek, RL Mercer, LR Bahl, and JK Baker. 1977. [Perplexity - measure of difficulty of speech recognition tasks](#). *Journal of the Acoustical Society of America*, 62(1):S63.
- Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. [Shakespeareizing modern language using copy-enriched sequence to sequence models](#). In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19, Copenhagen, Denmark. Association for Computational Linguistics.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L  lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Th  ophile Gervet, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2024. [Mixtral of experts](#). *Preprint*, arXiv:2401.04088.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. [How can we know what language models know?](#) *Transactions of the Association for Computational Linguistics*, 8:423–438.
- 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.
- Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. [Disentangled representation learning for non-parallel text style transfer](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy. Association for Computational Linguistics.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. [A convolutional neural network for modelling sentences](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.

- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *Preprint*, arXiv:2001.08361.
- Diederik P Kingma and Max Welling. 2022. [Auto-encoding variational bayes](#). *Preprint*, arXiv:1312.6114.
- Tom Kocmi and Christian Federmann. 2023. [Large language models are state-of-the-art evaluators of translation quality](#). In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.
- Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021a. [Generic resources are what you need: Style transfer tasks without task-specific parallel training data](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4241–4254, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021b. [Thank you BART! rewarding pre-trained models improves formality style transfer](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 484–494, Online. Association for Computational Linguistics.
- Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2023. [Multidimensional evaluation for text style transfer using chatgpt](#). *Preprint*, arXiv:2304.13462.
- Dongkyu Lee, Zhiliang Tian, Lanqing Xue, and Nevin L. Zhang. 2021. [Enhancing content preservation in text style transfer using reverse attention and conditional layer normalization](#). 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)*, pages 93–102, Online. Association for Computational Linguistics.
- Mina Lee, Katy Ilonka Gero, John Joon Young Chung, Simon Buckingham Shum, Vipul Raheja, Hua Shen, Subhashini Venugopalan, Thiemo Wambsganss, David Zhou, Emad A. Alghamdi, Tal August, Avinash Bhat, Madiha Zahrah Choksi, Senjuti Dutta, Jin L.C. Guo, Md Naimul Hoque, Yewon Kim, Simon Knight, Seyed Parsa Neshaei, Antonette Shibani, Disha Shrivastava, Lila Shroff, Agnia Sergeyuk, Jessi Stark, Sarah Stermann, Sitong Wang, Antoine Bosselut, Daniel Buschek, Joseph Chee Chang, Sherol Chen, Max Kreminski, Joonsuk Park, Roy Pea, Eugenia Ha Rim Rho, Zejiang Shen, and Pao Siangliulue. 2024. [A design space for intelligent and interactive writing assistants](#). In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI ’24, New York, NY, USA. Association for Computing Machinery.

- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- 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, New Orleans, Louisiana. Association for Computational Linguistics.
- Xiangyang Li, Xiang Long, Yu Xia, and Sujian Li. 2022. [Low resource style transfer via domain adaptive meta learning](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3014–3026, Seattle, United States. Association for Computational Linguistics.
- Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2017. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30.
- Yuan Li, Chunyuan Li, Yizhe Zhang, Xiujuan Li, Guoqing Zheng, Lawrence Carin, and Jianfeng Gao. 2020. [Complementary auxiliary classifiers for label-conditional text generation](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8303–8310.
- Yi Liao, Lidong Bing, Piji Li, Shuming Shi, Wai Lam, and Tong Zhang. 2018. [QuaSE: Sequence editing under quantifiable guidance](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3855–3864, Brussels, Belgium. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. 2020. [Revision in continuous space: Unsupervised text style transfer without adversarial learning](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8376–8383.
- Qingyi Liu, Jinghui Qin, Wenxuan Ye, Hao Mou, Yuxuan He, and Keze Wang. 2024. Adaptive prompt routing for arbitrary text style transfer with pre-trained language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18689–18697.

- Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. [Multi-task deep neural networks for natural language understanding](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.
- Zhengyuan Liu and Nancy Chen. 2022. [Learning from bootstrapping and stepwise reinforcement reward: A semi-supervised framework for text style transfer](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2633–2648, Seattle, United States. Association for Computational Linguistics.
- Varvara Logacheva, Daryna Dementieva, Irina Krotova, Alena Fenogenova, Irina Nikishina, Tatiana Shavrina, and Alexander Panchenko. 2022. [A study on manual and automatic evaluation for text style transfer: The case of detoxification](#). In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 90–101, Dublin, Ireland. Association for Computational Linguistics.
- Steven Loria. 2018. textblob documentation. *Release 0.16, 2*.
- Guoqing Luo, Yu Han, Lili Mou, and Mauajama Firdaus. 2023. [Prompt-based editing for text style transfer](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5740–5750, Singapore. Association for Computational Linguistics.
- Yiwei Lyu, Paul Pu Liang, Hai Pham, Eduard Hovy, Barnabás Póczos, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2021. [StylePTB: A compositional benchmark for fine-grained controllable text style transfer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2116–2138, Online. Association for Computational Linguistics.
- Yiwei Lyu, Tiange Luo, Jiacheng Shi, Todd Hollon, and Honglak Lee. 2023. [Fine-grained text style transfer with diffusion-based language models](#). In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 65–74, Toronto, Canada. Association for Computational Linguistics.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). *Preprint*, arXiv:2303.17651.
- David D. McDonald and James D. Pustejovsky. 1985. [A computational theory of prose style for natural language generation](#). In *Second Conference of the European Chapter of the Association for Computational Linguistics*, Geneva, Switzerland. Association for Computational Linguistics.
- Meta. 2024. [Introducing meta llama 3: The most capable openly available llm to date](#).
- Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. [Recurrent neural network based language model](#). In *Interspeech 2010*, pages 1045–1048.

- Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. [Evaluating style transfer for text](#). 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)*, pages 495–504, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alejandro Mosquera and Paloma Moreda. 2011. [The use of metrics for measuring informality levels in web 2.0 texts](#). In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*.
- Jonas Mueller, David Gifford, and Tommi Jaakkola. 2017. [Sequence to better sequence: Continuous revision of combinatorial structures](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2536–2544. PMLR.
- Sourabrata Mukherjee, Mateusz Lango, Zdenek Kasner, and Ondrej Dušek. 2024a. [A survey of text style transfer: Applications and ethical implications](#). *Preprint*, arXiv:2407.16737.
- Sourabrata Mukherjee, Atul Kr. Ojha, Akanksha Bansal, Deepak Alok, John P. McCrae, and Ondrej Dusek. 2024b. [Multilingual text style transfer: Datasets & models for Indian languages](#). In *Proceedings of the 17th International Natural Language Generation Conference*, pages 494–522, Tokyo, Japan. Association for Computational Linguistics.
- Sourabrata Mukherjee, Atul Kr. Ojha, and Ondrej Dusek. 2024c. [Are large language models actually good at text style transfer?](#) In *Proceedings of the 17th International Natural Language Generation Conference*, pages 523–539, Tokyo, Japan. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. [JFLEG: A fluency corpus and benchmark for grammatical error correction](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.
- Nasim Nouri. 2022. [Text style transfer via optimal transport](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2532–2541, Seattle, United States. Association for Computational Linguistics.
- OpenAI. [Chatgpt](#) [online]. 2022.
- OpenAI. 2024. [Gpt-4 technical report](#).
- Phil Sidney Ostheimer, Mayank Kumar Nagda, Marius Kloft, and Sophie Fellenz. 2024. [Text style transfer evaluation using large language models](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15802–15822, Torino, Italia. ELRA and ICCL.

- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. [Automatically neutralizing subjective bias in text](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):480–489.
- Juan Manuel Pérez, Juan Carlos Giudici, and Franco Luque. 2021. [pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks](#). *Preprint*, arXiv:2106.09462.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *OpenAI blog* 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).
- Sudha Rao and Joel Tetreault. 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)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.
- 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, Dublin, Ireland. Association for Computational Linguistics.
- Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. [Automatically identifying words that can serve as labels for few-shot text classification](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021. [Few-shot text generation with natural language instructions](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Tim Schopf, Karim Arabi, and Florian Matthes. 2023. [Exploring the landscape of natural language processing research](#). In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 1034–1045, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, volume 30.
- Xu Sheng, Fumiyo Fukumoto, Jiyi Li, Go Kentaro, and Yoshimi Suzuki. 2023. [Learning disentangled meaning and style representations for positive text reframing](#). In *Proceedings of the 16th International Natural Language Generation Conference*, pages 424–430, Prague, Czechia. Association for Computational Linguistics.
- Huashan Sun, Yixiao Wu, Yizhe Yang, Yinghao Li, Jiawei Li, Yuhao Ye, and Yang Gao. 2024. [PSST: A benchmark for evaluation-driven text public-speaking style transfer](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8438–8471, Miami, Florida, USA. Association for Computational Linguistics.
- 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Gemma Team. 2024. [Gemma: Open models based on gemini research and technology](#). *Preprint*, arXiv:2403.08295.
- Alexey Tikhonov, Viacheslav Shibaev, Aleksander Nagaev, Aigul Nugmanova, and Ivan P. Yamshchikov. 2019. [Style transfer for texts: Retrain, report errors, compare with rewrites](#). 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 3936–3945, Hong Kong, China. Association for Computational Linguistics.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.

- A. M. Turing. 1950. [I.—computing machinery and intelligence](#). *Mind*, LIX(236):433–460.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Shaolei Wang, Wangxiang Che, Qi Liu, Pengda Qin, Ting Liu, and William Yang Wang. 2020a. [Multi-task self-supervised learning for disfluency detection](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:9193–9200.
- Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. [Harnessing pre-trained neural networks with rules for formality 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 3573–3578, Hong Kong, China. Association for Computational Linguistics.
- Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and WenHan Chao. 2020b. [Formality style transfer with shared latent space](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2236–2249, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Daimeng Wei, Zhanglin Wu, Hengchao Shang, Zongyao Li, Minghan Wang, Jiaxin Guo, Xiaoyu Chen, Zhengzhe Yu, and Hao Yang. 2023. [Text style transfer back-translation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7944–7959, Toronto, Canada. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA. Curran Associates Inc.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.
- Zhang Xiang, Zhao Junbo, and LeCun Yann. 2015. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 28:649–657.
- Ruochen Xu, Tao Ge, and Furu Wei. 2019. [Formality style transfer with hybrid textual annotations](#). *Preprint*, arXiv:1903.06353.

- Sheng Xu, Yoshimi Suzuki, Jiyi Li, and Fumiyo Fukumoto. 2024. Decoupling style from contents for positive text reframing. In *Neural Information Processing*, pages 73–84, Singapore. Springer Nature Singapore.
- Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. [Paraphrasing for style](#). In *Proceedings of COLING 2012*, pages 2899–2914, Mumbai, India. The COLING 2012 Organizing Committee.
- Cheng Yang, Maosong Sun, Xiaoyuan Yi, and Wenhao Li. 2018a. [Stylistic Chinese poetry generation via unsupervised style disentanglement](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3960–3969, Brussels, Belgium. Association for Computational Linguistics.
- Heng Yang, Chen Zhang, and Ke Li. 2023a. [Pyabsa: A modularized framework for reproducible aspect-based sentiment analysis](#). In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 5117–5122.
- Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2023b. [Tailor: A soft-prompt-based approach to attribute-based controlled text generation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 410–427, Toronto, Canada. Association for Computational Linguistics.
- Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018b. [Unsupervised neural machine translation with weight sharing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–55, Melbourne, Australia. Association for Computational Linguistics.
- Catherine Yeh, Gonzalo Ramos, Rachel Ng, Andy Huntington, and Richard Banks. 2024. [Ghostwriter: Augmenting collaborative human-ai writing experiences through personalization and agency](#). *Preprint*, arXiv:2402.08855.
- Xiaoyuan Yi, Zhenghao Liu, Wenhao Li, and Maosong Sun. 2020. [Text style transfer via learning style instance supported latent space](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3801–3807. International Joint Conferences on Artificial Intelligence Organization. Main track.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. [Bertscore: Evaluating text generation with bert](#). *Preprint*, arXiv:1904.09675.
- Yi Zhang, Tao Ge, and Xu Sun. 2020b. [Parallel data augmentation for formality style transfer](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3221–3228, Online. Association for Computational Linguistics.
- Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. [Adversarially regularized autoencoders](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5902–5911. PMLR.

- Wanrong Zhu, Xin Wang, Tsu-Jui Fu, An Yan, Pradyumna Narayana, Kazoo Sone, Sugato Basu, and William Yang Wang. 2021. [Multimodal text style transfer for outdoor vision-and-language navigation](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1207–1221, Online. Association for Computational Linguistics.
- Xuekai Zhu, Jian Guan, Minlie Huang, and Juan Liu. 2023. [StoryTrans: Non-parallel story author-style transfer with discourse representations and content enhancing](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14803–14819, Toronto, Canada. Association for Computational Linguistics.
- Caleb Ziems, Minzhi Li, Anthony Zhang, and Diyi Yang. 2022. [Inducing positive perspectives with text reframing](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3682–3700, Dublin, Ireland. Association for Computational Linguistics.

Appendix

A Formalization of Reduction-and-Synthesis

Given the source text X , the expected inference Y with the target style s , we assume that a neutral text C sharing the same semantic information with X entails the style-free content which is preserved during transferring from X to Y . The SST task can be further decomposed as Eq. 1:

$$\begin{aligned}\mathbb{P}(Y|X, s) &= \frac{\mathbb{P}(Y, X, s)}{\mathbb{P}(X, s)} \\ &\geq \frac{\mathbb{P}(Y, X, C, s)}{\mathbb{P}(X, s)} \\ &= \frac{\mathbb{P}(Y, X, C, s)}{\mathbb{P}(X) \mathbb{P}(s)} \\ &= \frac{\mathbb{P}(X, C)}{\mathbb{P}(X)} \cdot \frac{\mathbb{P}(Y, X, C, s)}{\mathbb{P}(X, C) \mathbb{P}(s)} \\ &= \frac{\mathbb{P}(X, C)}{\mathbb{P}(X)} \cdot \frac{\mathbb{P}(Y, X, C, s)}{\mathbb{P}(X, C, s)} \\ &= \underbrace{\frac{\mathbb{P}(C|X)}{}}_{\text{reduction}} \underbrace{\frac{\mathbb{P}(Y|X, C, s)}{}}_{\text{synthesis}}\end{aligned}\tag{1}$$

B Hyperparameter

Hyperparameter	Value			
	ST	PG	PG2ST	ST2PG
Batch size	32	16	16	16
Number of epochs	2	3	3	2
Learning rate	$7e-5$	$4e-5$	$3e-5$	$4e-5$

Table 1: Hyperparameter settings of each variant for the experiment in Chapter 3.

Hyperparameter	Value		
	FDSC	w/o decoup	w/o contr
Batch size	16	16	32
Number of epochs	4	3	3
Learning rate	$7.2e - 5$	$6e - 5$	$7e - 5$

Table 2: Hyperparameter settings for each variant in the ablation study of Chapter 4. Hyperparameter settings of each variant during ablation study in Chapter 4. “w/o decoup” refers to removing the module on decoupling style from the content (training loss \mathcal{L}_{dsc}), and “w/o contr” indicates removing the contrastive learning module employed for preserving invariant contents (training loss \mathcal{L}_{contr})

C Prompt Templates

The prompt templates discussed in Chapter 5 primarily include three types, i.e., three types: generation, feedback, and refinement. These templates are employed at each inference step of the Self-Refine baseline, as well as in our reduction and synthesis pipeline. We illustrate these templates by employing an inference example on the Yelp dataset for the transfer task $neg \rightarrow pos$, as shown in Figures 1 ~ 9. Each of these templates is constructed with two instructions. Figures.1, 2, and 3 indicates the Self-Refine baseline. Figures.4, 5, and 6 refer to reduction phase, and Figures.7, 8, and 9 shows synthesis phase.

As introduced in Chapter 6, Figure 10, 11, and 12 illustrate the prompt templates for LLM-based evaluation using the SST task ($neg \rightarrow pos$) as an example. The $[input]$, $[reference]$, and $[generation]$ in each figure represent the placeholders for each input sentence, its corresponding annotated reference, and the output generated by LLM, respectively.

```

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to express the content with positive emotions.
Rewrite: I went to the restaurant and ate some chicken, it is delicious.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to express the content with positive emotions.
Rewrite: Salads are a delicious way to begin the meal.
###
Text: [input]
Rewrite the text to express the content with positive emotions.
Rewrite:

```

Figure 1: The generation prompt of the Self-Refine baseline. The `[input]` represents the placeholder for each input sentence.

```

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to express the content with positive emotions.
Rewrite: I went to the restaurant and ate some chicken.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just express the same content without positive emotions.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to express the content with positive emotions.
Rewrite: Salads are an appropriate way to begin the meal.
Does this rewrite meet the requirements?
Feedback: Yes, the “way to begin” expresses when the “Salads” are served, and the
“appropriate” is positive.
### a-b
Text: [input]
Rewrite the text to express the content with positive emotions.
Rewrite: [generation]
Does this rewrite meet the requirements?
Feedback:

```

Figure 2: The feedback prompt of the Self-Refine baseline. The `[input]` and `[generation]` represent the placeholders for each input sentence and the corresponding initial generation inferred by LLM, respectively.

```

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to express the content with positive emotions.
Rewrite: I went to the restaurant and ate some chicken.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just express the same content without positive emotions.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: I ate some noodles in this restaurant, it is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite does not mention the taste of "chicken" which is the topic of the text.
Rewrite: I went to the restaurant and ate some chicken, it is delicious.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to express the content with positive emotions.
Rewrite: Two staffs are serving for me, they are kind.
Does this rewrite meet the requirements?
Feedback: No, the "staffs are serving" is different from the topic about the taste of "Salads".
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: Salads are an inappropriate way to begin the meal.
Does this rewrite meet the requirements?
Feedback: No, the "way to begin" expresses when the "Salads" are served, but the "inappropriate" is still negative.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: Salads are an appropriate way to begin the meal.
###
Text: [input]
Rewrite the text to express the content with positive emotions.
Rewrite: [generation]
Does this rewrite meet the requirements?
Feedback: [feedback]
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite:

```

Figure 3: The refinement prompt of the Self-Refine baseline. The *[input]*, *[generation]*, and *[feedback]* represent the placeholders for each input sentence, its corresponding latest generation, and feedback inferred by LLM, respectively.


```

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: I went to the restaurant and ate some chicken.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: Salads are served to begin the meal.
###
Text: [input]
Rewrite the text to just explain the situation without any negative emotions.
Rewrite:

```

Figure 4: The generation prompt at the Reduction phase. The *[input]* represents the placeholder for each input sentence.

```

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: The chicken I ordered in this restaurant is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just duplicates the negative text, and “tasteless” represents
negative sentiment.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: Salads are served to begin the meal.
Does this rewrite meet the requirements?
Feedback: Yes, the rewrite expresses the content neutrally.
###
Text: [input]
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: [generation]
Does this rewrite meet the requirements?
Feedback:

```

Figure 5: The feedback prompt at the reduction phase. The *[input]* and *[generation]* represent the placeholders for each input sentence and the corresponding initial reduction inferred by LLM, respectively.

```

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: The chicken I ordered in this restaurant is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just duplicates the negative text, and “tasteless” represents
negative sentiment.
Okay, let’s try again. Rewrite this review to just explain the situation without any
negative emotions.
Rewrite: The chicken of the restaurant is not fresh.
Does this rewrite meet the requirements?
Feedback: No, the “chicken of the restaurant” express the same topic, but the “not
fresh” is still negative.
Okay, let’s try again. Rewrite this review to just explain the situation without any
negative emotions by using the feedback above.
Rewrite: I went to the restaurant and ate some chicken.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: Two staffs are serving for me.
Does this rewrite meet the requirements?
Feedback: No, the “staffs are serving” is different from the topic about the “Salads”.
Okay, let’s try again. Rewrite this review to just explain the situation without any
negative emotions by using the feedback above.
Rewrite: Salads are served.
Does this rewrite meet the requirements?
Feedback: No, the rewrite is the same topic about “salads” but it does not mention
when the “salads” are served.
Okay, let’s try again. Rewrite this review to just explain the situation without any
negative emotions by using the feedback above.
Rewrite: Salads are served to begin the meal.
###
Text: [input]
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: [generation]
Does this rewrite meet the requirements?
Feedback: [feedback]
Okay, let’s try again. Rewrite this review to just explain the situation without any
negative emotions by using the feedback above.
Rewrite:

```

Figure 6: The refinement prompt at the reduction phase. The *[input]*, *[generation]*, and *[feedback]* represent the placeholders for each input sentence, its corresponding latest reduction, and feedback inferred by LLM, respectively.

```

###
Text: The chicken I ordered in this restaurant is tasteless.
Content of the text: I went to the restaurant and ate some chicken.
Rewrite the text to express the content with positive emotions.
Rewrite: I went to the restaurant and ate some chicken, it is delicious.
###
Text: Salads are inappropriate for appetizers.
Content of the text: Salads are served to begin the meal.
Rewrite the text to express the content with positive emotions.
Rewrite: Salads are a delicious way to begin the meal.
###
Text: [input]
Content of the text: [reduction]
Rewrite the text to express the content with positive emotions.
Rewrite:

```

Figure 7: The generation prompt at the synthesis phase. The *[input]* represents the placeholder for each input sentence. The *[reduction]* is the final self-refined output from the reduction phase.

```

###
Text: The chicken I ordered in this restaurant is tasteless.
Content of the text: I went to the restaurant and ate some chicken.
Rewrite the text to express the content with positive emotions.
Rewrite: I ate some noodles in this restaurant, it is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite does not mention the taste of “chicken” which is the topic of the text.
###
Text: Salads are inappropriate for appetizers.
Content of the text: Salads are served to begin the meal.
Rewrite the text to express the content with positive emotions.
Rewrite: Salads are a delicious way to begin the meal.
Does this rewrite meet the requirements?
Feedback: Yes, the rewrite expresses when the “Salads” are served, the “they are delicious” are positive.
###
Text: [input]
Content of the text: [reduction]
Rewrite the text to express the content with positive emotions.
Rewrite: [generation].
Does this rewrite meet the requirements?
Feedback:

```

Figure 8: The feedback prompt at the synthesis phase. The *[input]* and *[generation]* represent the placeholders for each input sentence and its corresponding latest synthesis by LLM, respectively. The *[reduction]* is the final self-refined output from the reduction phase.

```

###
Text: The chicken I ordered in this restaurant is tasteless.
Content of the text: I went to the restaurant and ate some chicken.
Rewrite the text to express the content with positive emotions.
Rewrite: I ate some chicken in this restaurant.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just expresses the same content without positive emotions.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: I ate some noodles in this restaurant, it is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite does not mention the taste of "chicken" which is the topic of the text.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: I ate some chicken in this restaurant, it is tasteless.
###
Text: Salads are inappropriate for appetizers.
Content of the text: Salads are served to begin the meal.
Rewrite the text to express the content with positive emotions.
Rewrite: Two staff are serving for me, they are kind.
Does this rewrite meet the requirements?
Feedback: No, the "staff are serving" is different from the topic about the "Salads", although the "kind" is positive.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: Salads are delicious.
Does this rewrite meet the requirements?
Feedback: No, the rewrite is the same topic about "salads", but it does not mention when the "salads" are served.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: Salads are an appropriate way to begin the meal.
###
Text: [input]
Content of the text: [reduction]
Rewrite the text to express the content with positive emotions.
Rewrite: [generation]
Does this rewrite meet the requirements?
Feedback: [feedback]
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite:

```

Figure 9: The refinement prompt at the synthesis phase. The *[input]*, *[generation]*, and *[feedback]* represent the placeholders for each input sentence, its corresponding latest synthesis, and feedback inferred by LLM, respectively. The *[reduction]* is the final self-refined output from the reduction phase.

system: You are a helpful assistant for evaluating the sentiment style transfer task. The definition of this task is to revise the input sentence to transfer negative content to positive while preserving the sentiment-independent content.

user: Evaluate the following transfer case relative to the human reference on a continuous scale ranging from 0 to 100 points. A score of 0 indicates “no sentiment transferred” while a score of 100 denotes “perfect sentiment transferred”.

input sentence: *[input]*
human reference: *[reference]*
revised sentence: *[generation]*
Please only reply me the score.

Figure 10: Prompt template for evaluating the sentiment transfer strength.

system: You are a helpful assistant for evaluating the sentiment style transfer task. The definition of this task is to revise the input sentence to transfer negative content to positive while preserving the sentiment-independent content.

user: Evaluate the following transfer case relative to the human reference on a continuous scale ranging from 0 to 100 points. A score of 0 indicates “no preservation of sentiment-independent content” while a score of 100 denotes “perfect preservation of sentiment-independent content”.

input sentence: *[input]*
human reference: *[reference]*
revised sentence: *[generation]*
Please only reply me the score.

Figure 11: Prompt template for evaluating the capacity of preserving content.

system: You are a helpful assistant for evaluating the sentiment style transfer task. The definition of this task is to revise the input sentence to transfer negative content to positive while preserving the sentiment-independent content.

user: Evaluate the following transfer case relative to the human reference on a continuous scale ranging from 0 to 100 points. A score of 0 indicates “not fluent” while a score of 100 denotes “quite fluent”.

input sentence: *[input]*
human reference: *[reference]*
revised sentence: *[generation]*
Please only reply me the score.

Figure 12: Prompt template for evaluating the fluency.

D Demonstration

Figure 13 and 14 illustrated two transfer examples by using the disentangle pipeline discussed in Chapter 7. This demonstration is built based on “Open WebUI¹”, a self-hosted WebUI designed for offline interaction with local LLMs such as models loaded with Ollama.

Here is a sentence "ever since joes has changed hands it's just gotten worse and worse." Please analyze which part expresses negative sentiment, and which is sentiment-independent content.

OI llama3.2:latest

In the given sentence, "Ever since Joe's has changed hands, it's just gotten worse and worse," we can break down the components as follows:

1. "Ever since" - This phrase is a neutral expression indicating time, past or present. It doesn't convey any sentiment.
2. "Joe's has changed hands" - This part of the sentence expresses a fact or an event that occurred in the past. The word "changed hands" implies a change in ownership or management, which is a neutral event. The sentiment expressed here is not necessarily positive or negative.
3. "it's just gotten worse and worse" - This part of the sentence expresses negative sentiment. The use of the adverb "just" implies that the situation has been deteriorating recently, and the repetition of the word "worse" emphasizes a sense of increasing decline.

So, to summarize:

- "Ever since" is sentiment-independent content.
- "Joe's has changed hands" is also sentiment-independent content.
- "it's just gotten worse and worse" expresses negative sentiment.

Based on the analysis, please revise the sentence to transfer negative content to positive while preserving the sentiment-independent content.

OI llama3.2:latest

Here's a revised version of the sentence:

"Ever since Joe's has changed hands, it's just gotten better and better."

In this revised sentence, I've replaced the word "worse" with its opposite, "better", to transfer the negative sentiment to a positive one while preserving the sentiment-independent content. The phrase "it's just gotten [adjective]" is still neutral, but now it describes an improvement instead of a decline.

Note that this revision may not be suitable for every context or audience, as the original sentence had a more nuanced tone and implied a sense of disappointment or frustration. However, if you want to reframe the sentiment entirely, this revised sentence could work in some situations.

Figure 13: Disentanglement CoT by using LLaMA3.2 to transfer the sentence, “*ever since joes has changed hands it’s just gotten worse and worse.*”.

¹<https://openwebui.com/>

Here is a sentence "it isn't terrible, but it isn't very good either." Please analyze which part expresses negative sentiment, and which is sentiment-independent content.

OI llama3.2:latest

Let's break down the sentence:

"It isn't terrible"

- The word "terrible" has a strong negative connotation, implying something that is extremely bad or unpleasant.
- By saying "it isn't", you're using negation to convey that the thing or entity in question might be bad, but not as bad as being absolutely terrible.

The phrase "but it isn't very good either"

- The word "very" intensifies the adjective "good", making it an absolute statement: something is not just bad, but also extremely poor or of very low quality.
- The word "good" itself has a positive connotation, implying something that is desirable or satisfactory.

The part that expresses negative sentiment in this sentence is:

- "very good"

This phrase implies that the thing or entity in question is of a lower quality than expected, which is still a negative assessment. The use of the intensifier "very" amplifies the negative connotation of "good".

The sentiment-independent content in this sentence is:

- "It"

This word refers to a thing or entity without expressing any opinion or emotion about it. The sentence would still make sense even if the word "it" was replaced with a different noun.

Note that the overall tone of the sentence is ambivalent, as both parts convey negative assessments. However, one part (the first clause) implies a slight criticism, while the second part amplifies that criticism by highlighting the lack of quality.

Based on the analysis, please revise the sentence to transfer negative content to positive while preserving the sentiment-independent content.

OI llama3.2:latest

Here's a revised version of the sentence:

"It is excellent, but it meets expectations."

In this revised sentence, I've transferred the negative sentiments expressed in the original sentence to positive ones:

- Instead of saying "it isn't terrible", which implies that something might be bad or unpleasant, I've replaced it with "it is excellent". This phrase conveys a strong positive assessment.
- Instead of saying "but it isn't very good either", which implies that something is of poor quality, I've replaced it with "but it meets expectations". This phrase implies that the thing or entity in question is adequate or satisfactory, rather than exceeding expectations.

The only part that remains sentiment-independent content in this revised sentence is:

- "It"

This word still refers to a thing or entity without expressing any opinion or emotion about it. The sentence would still make sense even if the word "it" was replaced with a different noun.

Note that I've preserved the structure and tone of the original sentence, while flipping the negative sentiments into positive ones.

Figure 14: Disentanglement CoT by using LLaMA3.2 to transfer the sentence, *“it isn’t terrible, but it isn’t very good either.”*.