Prompt-Based Editing for Text Attribute Transfer

by

Guoqing Luo

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

Text Attribute Transfer (TAT) is a natural language processing task that involves changing certain attributes (e.g., sentiment and formality) of a given text while preserving other attributes. Recently, prompting approaches have been explored in TAT with the emergence of various pretrained language models (PLMs), where a textual prompt is used to query a PLM to generate attribute-transferred texts word by word in an autoregressive manner. However, such a generation process is less controllable and early prediction errors may negatively affect future word predictions. Consequently, these issues will lead to low performance in general.

In this thesis, we propose a prompt-based editing approach to text attribute transfer. Specifically, we prompt a PLM for text attribute classification and use the classification probability to derive a score of the attribute to be transferred. Then, we perform discrete search with word-level editing to maximize a comprehensive scoring function for a TAT task. In this way, we transform a prompt-based generation problem into a classification one, which does not suffer from the error accumulation problem and is more controllable than the autoregressive generation of sentences. In our experiments, we perform automatic evaluation on multiple benchmark datasets, and we show that our approach largely outperforms the existing systems that have 20 times more parameters. Additional empirical analyses further demonstrate the effectiveness of our approach.

Simplicity, is the essence of life.

Life is full of trial and error. One failure doesn't mean you're out of the picture.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Dr. Lili Mou. He is an admirable and careful researcher who has not only largely inspired me to progress on research projects but also encouraged me to do critical thinking and high-quality scientific writing.

As a supervisor, Lili is always patient and teaches me how to do research from scratch. In the past two years, we have spent countless hours together discussing research reports, math, and paper writing. During the project of prompt-based editing, we faced many difficulties. However, Lili contributed many great ideas, and we solved them one after another after a half-year exploration. Apart from research, Lili always shares stories with everyone, which greatly helps me broaden my horizon to the global world. Overall, Lili is a caring, diligent, and intelligent research mentor to me.

Further, I would like to thank three co-authors, Yuqiao Wen, for close collaboration and publishing the paper "An Empirical Study on the Overlapping Problem of Open-Domain Dialogue Datasets" in LREC 2022; and Yu Tong Han and Mauajama Firdaus for collaborating on the paper "Prompt-Based Editing for Text Style Transfer", which is accepted to EMNLP 20203 this year. Yuqiao Wen is a very friendly and responsible person. In our academic collaboration, he always listens carefully to my thoughts and guides positive discussions, which always makes our experiments proceed smoothly. In addition, Yu Tong Han and Mauajama Firdaus are also very helpful. They help run numerous experiments and carefully revise and proofread the manuscript. Without their efforts, I could not achieve what I have today.

Last but not least, I would also like to thank my parents, who have supported me over the past two years with countless remote conversations. They always listen to me carefully and encourage me to maintain a positive attitude towards life, as well as to bravely cope with difficulties in my research studies. I would not be who I am now without their unconditional support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

There has been a surge of interest in Natural Language Processing (NLP) in both industry and academia, largely driven by significant progress in the field of language models in recent years. This progress can be credited to a variety of factors, including access to more computational resources, datasets with better quality, and the development of deep learning architectures. As a result, NLP has reached a point where it is able to solve various complicated tasks such as machine translation (Sennrich et al., 2016; Lample et al., 2018a; Xu et al., 2021; Deguchi et al., 2023) and mathematical reasoning (Wei et al., 2022c; Wang et al., 2022; Kojima et al., 2022; Zhou et al., 2023).

In this thesis, we focus on text attribute transfer (Fu et al., 2019), a common research area of NLP, which aims to automatically rewrite a sentence, changing certain attributes from one type to another, such as transferring the positive-sentiment sentence "He loves playing different sports" into a negative one "He hates playing different sports". An example is illustrated in Figure 1.1. During the transfer, the designated attributes (e.g., sentiment and formality) of a sentence must be changed, whereas other attributes should be preserved. Text attribute transfer (TAT) has wide applications in the real world, such as personalized response generation (Yang et al., 2017; Zheng et al., 2021), text debiasing (Nogueira dos Santos et al., 2018; Xiang et al., 2012; Ma et al., 2020), text simplification (Woodsend and Lapata, 2011; Dong

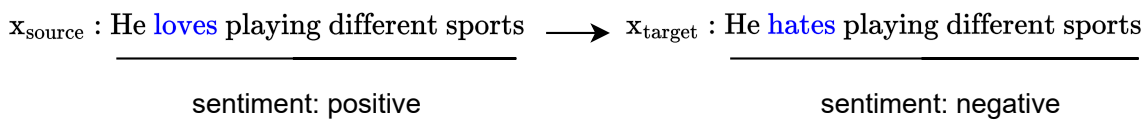

Figure 1.1: Example of text attribute transfer. A positive-sentiment sentence is transferred to be a negative one.

et al., 2019; Kumar et al., 2020), paraphrase generation (Chen et al., 2019; Liu et al., 2020b; Krishna et al., 2020), and stylistic headline generation (Liu et al., 2020a; Jin et al., 2020; Zhan et al., 2022).

Early work on TAT mainly falls into three categories, and each of them has its own drawback:

- **Parallel supervision** with labelled source–target sentence pairs in a sequence-to-sequence manner (Zhu et al., 2010; Rao and Tetreault, 2018; Zhang et al., 2020b). However, obtaining parallel training data is labor-intensive and time-consuming, which remains a significant challenge for TAT.

- **Non-parallel supervision** with labels only, such as learning latent representations of sentences (Shen et al., 2017; John et al., 2019; Goyal et al., 2021). However, for some sentences, the representation of those designated attributes and the other attributes are implicitly intertwined, and the explicit separation of them is not always possible, resulting in suboptimal model performance.

- **Unsupervised generative methods**, such as constructing pseudo-parallel training data for learning (Lample et al., 2018b; Luo et al., 2019; Krishna et al., 2020). However, these methods require a complicated training process to optimize the model, which lacks efficiency. In addition, poor-quality data construction would also possibly lead to low performance in general.

Very recently, prompting methods have been explored in TAT (Reif et al., 2022; Suzgun et al., 2022), as large-scale pretrained language models (PLMs) enable us to perform various natural language generation tasks in a zero-shot (Wei et al., 2022a;

Sanh et al., 2022; Kojima et al., 2022) or exemplar-based manner (Brown et al., 2020; Schick and Schütze, 2021a; Wei et al., 2022c).

Previous work uses a prompt (e.g., a piece of text "Rewrite the text to be positive:") to query a PLM, which will then generate an attribute-transferred sentence in an autoregressive manner (Reif et al., 2022; Suzgun et al., 2022). However, such autoregressive generation is less controllable, as words are generated one after another by the PLM. This can lead to an error accumulation issue where early prediction errors of the PLM will affect its future predictions, leading to less satisfactory performance in general. In this thesis, we also follow the prompt-based setting. This setting does not require any training samples or labels, but directly performs inference with PLMs; thus, it is more challenging than the above three settings.

1.2 Thesis Contributions

In this thesis, we propose a prompt-based editing approach to TAT, aiming to make the attribute-transfer generation more controllable. We first design a PLM-based classifier. Specifically, we prompt a PLM for classification of the attribute to be transferred and use the classification probability to compute a score of the attribute to be transferred, denoted as f_{transfer}. Then, we perform steepest-ascent hill climbing (SAHC; Russell and Norvig, 2010) for discrete search with word-level editing (such as replacement, insertion, and deletion) to maximize a heuristically defined scoring function for TAT tasks. In this way, we transform a prompt-based generation problem into a classification one, which mainly involves a one-word prediction and is generally believed to be easier than multiple-word predictions in an autoregressive sentence generation.

Our approach provides several advantages. First, it does not suffer from the error accumulation problem, because it performs word-level local edits scattered throughout the entire sentence rather than generating a sentence word by word. Further, we design a discrete search algorithm through iterative editing. This algorithm combines

the score derived from attribute classification with other scoring functions, including language fluency and non-transfer attribute similarity, and leads to a more controllable and refined generation of sentences.

In our experiments, we use Eleuther AI's GPT-J-6B (an off-the-shelf PLM)[1] and conduct automatic evaluation on multiple benchmark datasets. The experimental results show that our prompt-based editing approach largely outperforms the existing prompting systems that have 20 times more parameters. Additional empirical analysis shows the effectiveness of different scoring components and the proposed discrete search algorithm in our approach.

To sum up, the main contributions of this paper include:

- We propose a prompt-based editing approach, which transforms a prompt-based text generation into a classification problem on text attribute transfer, and generates more controllable sentences than autoregressive generation.
- We design a discrete search algorithm for editing, further ensuring the controllable generation of sentences.
- We conduct comprehensive experiments and provide detailed empirical analyses on multiple benchmark datasets to verify the effectiveness of our approach.

1.3 Thesis Structure

In this chapter, I introduced the background and motivation of this thesis, and stated the thesis contributions.

In Chapter 2, I will present the related work. In particular, I will start by providing the background for language models, ranging from explaining the basic Transformer architecture to introducing both regular-scale pretrained language models and large-scale language models. Then, I will introduce different prompting approaches and several paradigms of text attribute transfer.

[1] https://github.com/kingoflolz/mesh-transformer-jax

In Chapter 3, I will demonstrate the details of our prompt-based editing approach to text attribute transfer. I will first explain the framework of our prompt-based classifier, which is designed to transform the prompt-based generation problem into a classification one and also to compute a score of the attribute to be transferred. Second, I will introduce two additional search objectives that help measure the quality of a sentence. Then, I will introduce our proposed discrete search algorithm for iterative word-level editing.

In Chapter 4, I will present the experimental results of our approach. It starts with an introduction to several experimental setups. Then we will show the main results, followed by quantitative and qualitative analyses to demonstrate the effectiveness of different components in our approach.

In Chapter 5, I will conclude the findings and contributions of this thesis. We will also discuss the limitations of our approach that can be addressed in the future.

Chapter 2

Background and Related Work

2.1 Overview

Natural Language Processing (NLP) is a field within Artificial Intelligence (AI), with the goal of enabling machines to automatically comprehend and produce human-like texts. NLP involves two aspects: natural language understanding (NLU) and natural language generation (NLG), and our work focuses on the latter. In recent years, the development of pretrained language models (PLMs) has led to strong performance in various NLG tasks. Following this trend, our work delves into text attribute transfer (TAT), a subarea of NLG, aiming to transform certain attributes of a sentence from one type to another while preserving the other attributes.

In this chapter, I will introduce the background knowledge of our research work. In Section 2.2, I will describe the studies on language models from the Transformer architecture to different categories of PLMs. In Section 2.3, I will introduce two types of prompting methods: prompt-based classification and generation. In Section 2.4, I will explain several paradigms of TAT: parallel supervision, non-parallel supervision, and unsupervised learning methods, followed by an introduction to the recently proposed prompt-based paradigm.

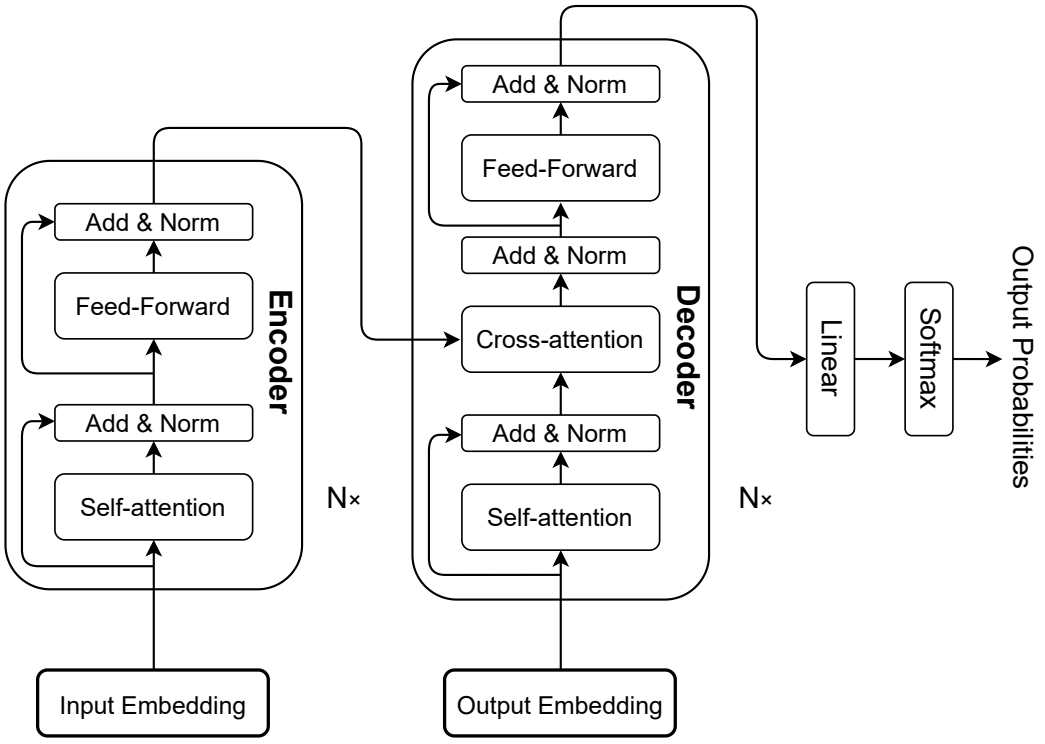

Figure 2.1: The Transformer architecture. This diagram is adapted from (Vaswani et al., 2017). Note that the self-attention layer in the decoder is slightly modified from the one in the encoder, and we omit the difference for simplicity.

2.2 Language Models

2.2.1 Transformer

The Transformer (Vaswani et al., 2017) is a prevailing neural architecture that is initially designed as a sequence-to-sequence model with an encoder and a decoder. In particular, the encoder encodes the input sequence into a list of continuous embeddings that incorporate information about the entire input; the decoder then decodes these embeddings for the next-token prediction.

The architecture of the Transformer is shown in Figure 2.1. Specifically, the encoder and decoder are composed of a stack of N layers with the same parameters. Each layer in the encoder consists of two sub-layers: a self-attention layer and a fully connected feed-forward neural network. In addition, residual connections (He et al., 2016) and layer normalization (Ba et al., 2016) are employed around these two sub-layers.

Different from the encoder, the decoder inserts an additional cross-attention layer, which is situated between the self-attention layer and the feed-forward neural network, acting on the encoder output. A linear transformation and softmax layer are further applied to convert the decoder output into a categorical probability distribution of the next token over the Transformer's entire vocabulary.

The key to the Transformer is the use of an attention mechanism, also called *Scaled Dot-Product Attention*, focusing on different parts of the input sequence when generating the output. The Transformer attention is computed under three matrix representations, which are queries Q, keys K, and values V. In particular, the attention is given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{2.1}$$

where $QK^\top$ is the dot product of queries with all the keys, d_k is the dimension of the queries and keys, and $\frac{1}{\sqrt{d_k}}$ is used as a scaling factor to mitigate the gradient vanishing problem caused by the softmax function (Vaswani et al., 2017).

2.2.2 Pretrained Language Models

Leveraging the Transformer architecture, researchers have developed various pretrained language models (PLMs), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), GPT-2 (Radford et al., 2019), and T5 (Raffel et al., 2020). These language models are pretrained on extensive datasets and consist of a massive number of parameters. They are then used as a backbone and achieve strong performance on various NLP tasks (Liu et al., 2021). Generally, these PLMs are developed into three categories:

- Encoder-only models, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), utilize the Transformer encoder architecture. When words from a piece of text are fed as input, the self-attention mechanism processes each word, incorporating both left and right context and thus rendering the attention

mechanism bidirectional. This feature enables the model to perform masked word prediction in many text generation tasks, where a word in a sentence is masked out and the model predicts the missing word based on the surrounding context.

- Decoder-only models, such as GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), are based on the Transformer decoder architecture. Different from encoder-only models, these decoder-only models use causal attention to process a piece of text, where the prediction of each word only depends on the preceding words in the sequence and not on future words. This allows the model to treat the input sequence as a prefix and then generate an output in a left-to-right manner. Consequently, these models are typically used for left-to-right sequence generation tasks.

- Encoder–decoder models, such as T5 (Raffel et al., 2020) and BART (Lewis et al., 2020), have two Transformer components: an encoder and a decoder. The encoder converts an input sequence into a contextual representation, and the decoder uses the representation to generate a corresponding output sequence. Therefore, these models are mainly applied to sequence-to-sequence text generation tasks.

Overall, PLMs in the encoder-only, decoder-only, and encoder–decoder categories each have distinct architectures and specific application scenarios, highlighting the adaptability of the Transformer architecture. Their unique features have driven advancements across various NLP tasks, ranging from masked word prediction to text generation, and lead to a notable trend of scaling up PLMs. In the following subsection, I will demonstrate the progress achieved by scaling up these PLMs.

2.2.3 Large Language Models

Recently, researchers have found that scaling up the model size, pretraining data size, and the amount of computation resource leads to a largely improved performance and capacity for a variety of tasks, following the scaling law in Kaplan et al. (2020). This phenomenon motivates the creation of ever-larger PLMs, such as the 175-billion-parameter GPT-3 (Brown et al., 2020), the 540-billion-parameter PaLM (Chowdhery et al., 2022), and the 176-billion-parameter BLOOM (Scao et al., 2022).

Although the scaling operation is mainly conducted on model size while maintaining a similar Transformer architecture and pretraining data, these large-scale PLMs show different behaviors from regular-scale PLMs (e.g., 330-million-parameter BERT and 1.5-billion-parameter GPT-2) and demonstrate surprising abilities to solve a series of challenging NLP tasks (Wei et al., 2022b). For example, GPT-3 is able to achieve strong performance on the commonsense, logical, and arithmetic reasoning tasks[1] with only a few exemplars in terms of in-context learning (Brown et al., 2020), whereas a 350-million-parameter GPT model performs poorly (Wei et al., 2022c). To this end, researchers use the term, *Large language models* (LLMs), to categorize these large-scale PLMs (Brown et al., 2020; Chowdhery et al., 2022; Scao et al., 2022; Thoppilan et al., 2022). Then, ChatGPT[2] (or GPT-3.5) and GPT-4 (OpenAI, 2023), both adapted from the basic GPT model, have exhibited an extraordinary ability to engage in conversations with humans and even multimodal capabilities of interacting with images, videos, speech, and code. Overall, LLMs are able to tackle complex tasks and deliver great performance in understanding and generating human-like texts, representing a monumental breakthrough in the field of NLP.

In this thesis, we use a regular-scale PLM, Eleuther AI's GPT-J-6B[3] in our approach and compare the experimental results with several LLMs, including 175-

[1]It should also be mentioned that only models that are over 100 billion parameters yield performance gains on these reasoning tasks (Wei et al., 2022c).

[2]https://openai.com/blog/chatgpt/

[3]https://github.com/kingoflolz/mesh-transformer-jax

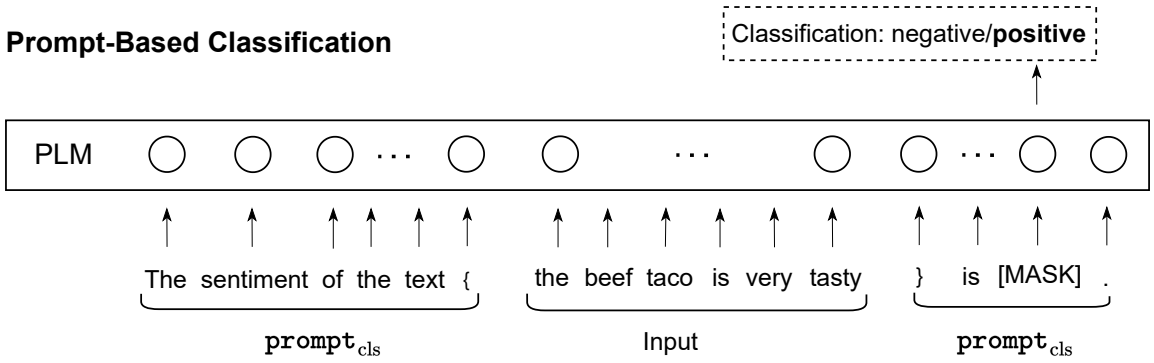

Figure 2.2: An overview of prompt-based classification. Previous work (Gao et al., 2021) uses a masked language model with a textual template to perform text classification.

billion-parameter GPT-3, 128-billion-parameter LLM, and 128-billion-parameter LLM-dialog, on two attribute transfer benchmark datasets.

2.3 Prompting Methods

Prompting methods use a piece of text to query a PLM to provide desired outputs (Liu et al., 2021). The simplest prompting method, perhaps, is zero-shot prompting (Wei et al., 2022a; Sanh et al., 2022; Suzgun et al., 2022), which directly prompts a PLM to perform an NLP task (see Figure 2.2 and 2.3), but may result in returning less well-formatted or illogical sentences (Reif et al., 2022). Another prompting method is few-shot prompting (Brown et al., 2020; Schick and Schütze, 2021a,b; Wei et al., 2022c). This method requires several task-specific exemplars for the PLMs, but it is able to achieve higher performance than zero-shot prompting and is therefore more widely applied in various NLP tasks (Schick and Schütze, 2021a; Brown et al., 2020; Wei et al., 2022c).

2.3.1 Prompt-Based Classification

Prompting methods were initially applied to natural language classification tasks such as sentiment classification, textual entailment, and question answering (Brown et al., 2020; Schick and Schütze, 2021b; Gao et al., 2021; Min et al., 2022; Wei et al., 2022a).

An overview of prompt-based classification is shown in Figure 2.2, where a PLM is asked to predict the masked word given a prompt "The sentiment of the text {*the beef taco is tasty*} is [MASK]." for binary sentiment classification, and the predicted word is then projected to a label *positive* by a pre-defined verbalizer (Schick and Schütze, 2021b). Such a prediction process is also called *masked language modeling* (Devlin et al., 2019).

Different from masked language modeling that predicts a word inside a sentence, another widely applied approach to prompt-based classification is next-word prediction. This approach mainly uses GPT model series, such as GPT-2 (Radford et al., 2019), to predict the next word conditioning on the whole preceding words in a sequence, denoted as

$$P(\mathrm{s}_{T+1}|\mathrm{s}_1, \cdots, \mathrm{s}_T) \tag{2.2}$$

where a prompt is modeled as an input sequence $\mathbf{s} = (\mathrm{s}_1, \mathrm{s}_2, \cdots, \mathrm{s}_T)$ with length T, and the next word s_{T+1} is predicted for classification.

2.3.2 Prompt-Based Generation

With the emergence of various PLMs (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020; Raffel et al., 2020), prompting methods have recently been widely applied to natural language generation tasks (Liu et al., 2021), such as text attribute transfer (Reif et al., 2022; Suzgun et al., 2022), machine translation (Radford et al., 2019; Brown et al., 2020; Raffel et al., 2020), and text summrization (Schumann et al., 2020; Liu et al., 2022a). An overview of prompt-based generation is shown in Figure 2.3, where a piece of text "Rewrite the sentence to be more positive: *the beef taco is bland*" prompts the PLM to perform next-token prediction multiple times and thus autoregressively generate a complete sentence *the beef taco is tasty*.

Recently, generative reasoning tasks, such as commonsense, symbolic, and arithmetic reasoning, have been built to explore LLMs' ability to solve complicated tasks (Rae et al., 2021; Srivastava et al., 2023). Concurrently, a series of chain-of-thoughts (CoT)

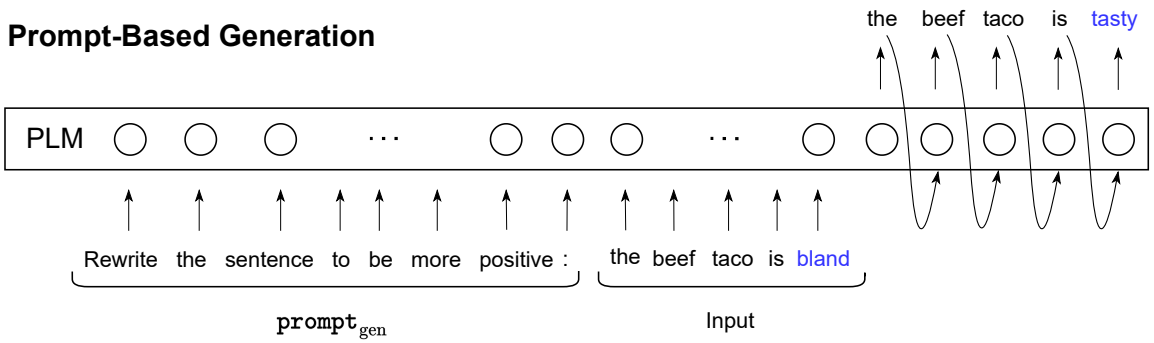

Figure 2.3: An illustration of prompt-based generation: previous work (Reif et al., 2022) uses a prompt to query a frozen pretrained language model (PLM), which generates an attribute-transferred sentence in an autoregressive manner.

prompting methods have been proposed to show LLMs' reasoning ability (Wei et al., 2022c; Wang et al., 2022; Zhou et al., 2023; Kojima et al., 2022; Li et al., 2023). Wei et al. (2022c) propose CoT prompting to solve some complicated generative reasoning tasks in a few-shot manner. This prompting method uses a few manually designed exemplars containing reasoning paths to encourage LLMs to perform multi-step thinking and then generate a reasoning path with the final answer for the question. Wang et al. (2022) generate a number of reasoning paths from an LLM via CoI prompting and use a majority voting mechanism to select the most consistent output as the answer. In addition, Kojima et al. (2022) propose zero-shot CoT prompting, which adds a piece of reasoning text *Let's think step by step* after the question to generate a reasoning path at the first stage, and then combines the generated path with the question to generate a final answer. Zhou et al. (2023) decompose a complex problem into multiple subproblems and solve them sequentially before obtaining the final answer. Overall, these prompting methods mainly propose different prompt engineering techniques to encourage LLMs to perform reasoning in a stepwise manner and generate desired answers.

Attribute	Source	Target
Politeness	**Polite:** Could you please send me the data?	**Impolite:** Send me the data!!
Simplicity	**Expert:** Many cause dyspnea, pleuritic chest pain.	**Layman:** The most common symptoms, regardless of the type of fluid in the pleural space or its cause, are shortness of breath and chest pain.
Biasedness	**Biased:** A new downtown is being developed which will bring back...	**Neutral:** A new downtown is being developed which its promoters hope will bring back...
Grammar	**Ungrammatical:** My cousin is 12years old.	**Grammatical:** My cousin is 12 years old.

Table 2.1: Illustrative examples of different attribute transfer tasks.

2.4 Text Attribute Transfer

Text attribute transfer (TAT) aims at transforming certain attributes (e.g., sentiment and formality) of a given text while keeping the other attributes (Fu et al., 2019). Other than the sentiment mentioned in Figure 1.1, there are also some other commonly explored attributes, such as politeness, simplicity, biasedness, and grammar. Examples of these attributes are illustrated in Table 2.1.

Traditional approaches to attribute-transfer generation involve supervised methods with parallel training data, where a source sentence is paired with a target sentence (Xu et al., 2012; Zhang et al., 2015; Rao and Tetreault, 2018; Wang et al., 2019, 2020). Xu et al. (2012) use a phrase-based machine translation model to explore automatic paraphrasing in a designated attribute and evaluate the effectiveness of new metrics as well as models in generating attribute-transferred paraphrases. Wang et al. (2019) explore three different strategies to incorporate rule-based systems into a language model, keeping more information from the original sentence. Wang et al. (2020) integrate rule-based systems with a language model to employ a shared encoder and two decoders to capture the attributes of formality and informality, respectively, and use auxiliary losses to ensure the shared latent space captures semantic information. However, obtaining parallel data is labor-intensive and time-consuming, remaining a significant challenge for this task.

To mitigate the need for parallel data, one line of research focuses on non-parallel

supervision, where it trains the model on a non-parallel but attribute-labeled corpus (Shen et al., 2017; Bao et al., 2019; Goyal et al., 2021; John et al., 2019; Li et al., 2018; Riley et al., 2021). John et al. (2019) train an autoencoder to disentangle the representation of designated attributes and other attributes in a sentence. Riley et al. (2021) extract a vector of the attribute to be transferred from a piece of text and use it to condition the decoder for denoising and reconstructing a corrupted sentence. Li et al. (2018) combine retrieval and generation to edit a candidate sentence similar to the source input incrementally. Goyal et al. (2021) train multiple language models as discriminators for each of the designated attributes. However, explicit separation of the representation of attributes to be transferred and other attributes is not always possible because in some sentences they can be implicitly intertwined.

Another line of research is devoted to unsupervised learning methods, which constructs pseudo-parallel training data for pretraining the model (Lample et al., 2018b; Luo et al., 2019; Chen et al., 2019; Krishna et al., 2020; Reid and Zhong, 2021). Luo et al. (2019) generate pseudo-parallel training data via back-translation (Lample et al., 2018a) and apply policy gradient training to learn one-step mappings between the corpora of source and target attributes. Krishna et al. (2020) create pseudo-parallel training data via two-step paraphrasing to fine-tune a language model for generating attribute-transferred sentences. Reid and Zhong (2021) first train an attribute classifier to perform synthesis of source–target sentence pairs, which are then used to train a Levenshtein editor and perform multi-span edits. However, these unsupervised learning methods require a complicated training procedure, which is not efficient. In addition, poor-quality data synthesis would possibly lead to low performance in general.

Recently, with the emergence of various LLMs, researchers have developed several prompt-based approaches that generate attribute-transferred texts in a zero-shot (Suzgun et al., 2022) or exemplar-based manner (Reif et al., 2022). Such methods do not require a learning process or any training labels. Reif et al. (2022) prompt

large-scale PLMs to generate sentences in various designated attributes. Suzgun et al. (2022) generate multiple candidate sentences and then use a re-ranking mechanism to choose one with the highest score as the final output.

Our approach follows the prompt-based setting and directly performs attribute transfer without any training procedure. However, unlike other work that mainly performs autoregressive generation, our approach proposes a new prompt-based editing paradigm for text generation, where we not only design a PLM-based scoring function but also develop a discrete search algorithm that is particularly suited to our scenario.

2.5 Summary

In this section, I first introduced the Transformer architecture and three categories of PLMs: encoder-only, decoder-only, and encoder–decoder models. I also discussed the recent development of LLMs, which have achieved strong performance in interacting with humans in conversations and also multimodal abilities of image, videos, speech, and code, attracting people's attention all over the world. Then, I moved on to the introduction of two different types of prompting methods, prompt-based classification and generation. They both use a piece of text to query a PLM to either perform the next-word prediction for classification or an autoregressive generation of words. These two prompting methods have been rapidly developed with the emergence of LLMs.

Finally, I discussed different categories of paradigms for text attribute transfer (TAT). I started with a traditional approach to this task, parallel supervision, which requires labor and time. Then I moved on to non-parallel supervision and unsupervised learning methods, which are utilized to alleviate the demand for parallel data. However, non-parallel supervision may not explicitly separate the representation of attributes to be transferred and other attributes in some sentences where the representation is intertwined; unsupervised learning methods may lack efficiency, and the synthesis of low-quality data could result in poor performance. I also introduced

the recently proposed prompt-based paradigm for the task of TAT, which utilizes the LLMs and does not require a training procedure. Our approach also follows the prompt-based paradigm.

Chapter 3

Approach

3.1 Overview

In this chapter, I will explain the details of our prompt-based editing approach to text attribute transfer. Given an input sentence $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_n)$, our goal is to generate a sentence $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, \cdots, \mathrm{y}_n)$ that transfers the designated attribute of $\mathbf{x}$. As shown in Figure 3.1, the framework of our approach involves prompting a pretrained language model (PLM) to predict the sentiment of a candidate sentence. Then, we perform discrete search and iterative word-level edits of a candidate sentence to maximize a comprehensive scoring function that involves the PLM's classification probability. The highest-scored candidate is taken as the attribute-transferred sentence.

The approach consists of three main components: a prompt-based classifier, search objective, and discrete search. In Section 3.2, I will introduce the prompt-based classifier, which is designed to perform attribute classification using a PLM and compute a score of the attribute to be transferred. In Section 3.3, I will explain two additional search objectives that help control the quality of generated sentences in our approach: language fluency and non-transfer attribute similarity. In Section 3.4, I will introduce three types of editing operations and the discrete search algorithm applied in our approach.

3.2 Prompt-Based Classifier

In previous work, researchers directly prompt a PLM to obtain attribute-transferred sentences (Figure 2.3; Reif et al., 2022; Suzgun et al., 2022). However, this could be a challenging process, as the PLM has to generate the sentence in a zero-shot or exemplar-based manner; such a process is autoregressive and less controllable.

To address this, we design a prompt-based classifier, transforming the attribute transfer task from text generation into a classification problem. Specifically, we prompt a PLM for attribute classification and calculate the score of the attribute to be transferred. This involves only one single-step prediction and is much simpler than generating the whole sentence.

Given a candidate sentence [$\mathbf{y}$], we intuitively design the prompt as

$$\texttt{prompt}_{\text{cls}}(\mathbf{y}) \equiv \text{The [t] of the text \{ [}\mathbf{y}\text{] \} is :} \tag{3.1}$$

where [t] is the attribute to be transferred, such as *sentiment* or *formality* in our experiments, and "{" and "}" are delimiter pairs. It should be mentioned that we follow Reif et al. (2022) and mainly use the delimiter pairs "{" and "}" in our prompt for experiments. More detail of prompt engineering is shown in Section 4.5.3.

Based on the above prompt, we perform the next-word prediction to obtain a probability. Specifically, the PLM computes the conditional probability of the next word w in the vocabulary given the prompt, denoted by $P_{\text{PLM}}(\text{w} \mid \texttt{prompt}_{\text{cls}}(\mathbf{y}))$.

Instead of looking at the probability distribution over the whole vocabulary of a PLM, we restrict our attention to the attribute to be transferred. We denote s_i as the representative word of the ith value of the attribute to be transferred. In our experiments, we have several specific settings: sentiment, formality, and Shakespeare-to-modern transfer. Here, s_i is simply chosen to be the most intuitive word, namely, *positive* and *negative* for the sentiment setting, *formal* and *informal* for the formality setting, *old* and *modern* for the Shakespeare-to-modern setting[1]. In gen-

[1]We followed the prompt in Suzgun et al. (2022) and used *old* to represent the attribute of

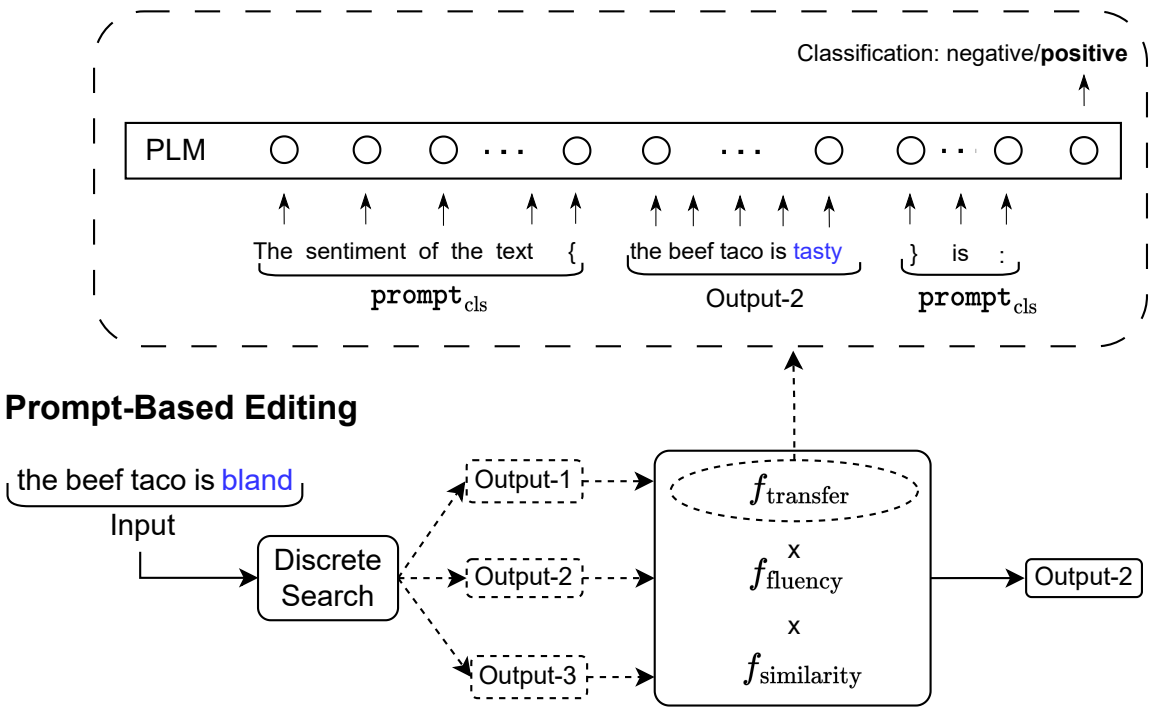

Figure 3.1: An illustration of our prompt-based editing approach, which involves one-word classification (e.g., *positive* or *negative* in sentiment transfer; *formal* or *informal* in formality transfer).

eral, the predicted probabilities of the two attributes are $P_{\text{PLM}}(\text{s}_1 \mid \texttt{prompt}_{\text{cls}}(\mathbf{y}))$ and $P_{\text{PLM}}(\text{s}_2 \mid \texttt{prompt}_{\text{cls}}(\mathbf{y}))$.

To compute the score of the attribute to be transferred, we consider the ratio of those two probabilities. Suppose a sentence in one transfer attribute s_1 is to be changed to another attribute s_2, we design the score as follows:

$$f_{\text{transfer}}(\mathbf{y}) = \frac{P_{\text{PLM}}(\text{s}_2 \mid \texttt{prompt}_{\text{cls}}(\mathbf{y}))}{P_{\text{PLM}}(\text{s}_1 \mid \texttt{prompt}_{\text{cls}}(\mathbf{y}))} \tag{3.2}$$

Such a ratio measures the candidate's relative affiliation with different attributes to be transferred.[2] It is more robust than the predicted probability $P_{\text{PLM}}(\cdot \mid \texttt{prompt}_{\text{cls}}(\mathbf{y}))$, which could be affected by the data sample *per se*.

Shakespeare.

[2]While our datasets only consider the transfer between two attributes, our approach can be easily extended to multiple attributes in a one-vs-one or one-vs-all manner.

3.3 Search Objective

We apply an edit-based search for text attribute transfer. This follows the recent development of search-based text generation (Li et al., 2020; Kumar et al., 2020; Jolly et al., 2022; Liu et al., 2022a; Mou, 2022), where local edits (e.g., word changes) are performed to maximize a heuristically defined objective function. However, different from previous search-based work, we propose to prompt an off-the-shelf PLM to compute the score $f_{\text{transfer}}(\mathbf{y})$ and do not require any task-specific training procedure.

Overall, our objective function involves three aspects:

$$f(\mathbf{y}; \mathbf{x}) = f_{\text{transfer}}(\mathbf{y}) \cdot f_{\text{fluency}}(\mathbf{y}) \cdot f_{\text{similarity}}(\mathbf{y}, \mathbf{x}) \tag{3.3}$$

where the scorer f_{transfer} is designed in Section 3.2; f_{fluency} and $f_{\text{similarity}}$ are language fluency and non-transfer attribute similarity scorers, mostly adopted from previous work and explained in detail below.

3.3.1 Language Fluency

A language fluency scorer provides an approximation of how fluent a candidate sentence $\mathbf{y}$ is. In our work, we follow Suzgun et al. (2022) and use GPT-2 (Radford et al., 2019) to obtain the fluency score. For a tokenized candidate output $\mathbf{y} = \{\mathrm{y}_1, \mathrm{y}_2, \cdots, \mathrm{y}_t\}$, we first calculate the perplexity (PPL) of $\mathbf{y}$:

$$\text{PPL}(\mathbf{y}) = \exp\left\{-\frac{1}{t}\sum_{i}^{t} \log P_{\text{GPT-2}}(\mathrm{y}_i|\mathbf{y}_{<i})\right\} \tag{3.4}$$

where $P_{\text{GPT-2}}(\mathrm{y}_i|\mathbf{y}_{<i})$ is the likelihood of the ith token y_i conditioned on the preceding tokens $\mathbf{y}_{<i}$ according to the GPT-2 model, and t is the length of $\mathbf{y}$. We use the entire vocabulary of the GPT-2 model with the widely used byte-pair encoding (BPE) algorithm (Sennrich et al., 2016), which encodes rare and unknown words as sequences of common subword units, and y_i here is a token after the BPE segmentation.

Since a lower PPL means a more fluent sequence, we use the reciprocal of PPL to

represent the fluency score:

$$f_{\text{fluency}}(\mathbf{y}) = \left[\frac{1}{\text{PPL}(\mathbf{y})}\right]^{\alpha} \tag{3.5}$$

By substituting Equation 3.4 into Equation 3.5, we obtain the fluency score of the candidate output $\mathbf{y}$:

$$f_{\text{fluency}}(\mathbf{y}) = \left(\left[\prod_{i=1}^{t} P_{\text{GPT-2}}(\text{y}_i|\mathbf{y}_{<i})\right]^{\frac{1}{t}}\right)^{\alpha} \tag{3.6}$$

where α is a hyperparameter balancing f_{fluency} with other scoring functions. Notice that a weighting hyperparameter is not needed for the scorer f_{transfer} because the relative weights of different scorers are given in f_{fluency} and $f_{\text{similarity}}$.

3.3.2 Non-Transfer Attribute Similarity

The non-transfer attribute similarity scorer measures the similarity of attributes to be preserved between a candidate sentence $\mathbf{y}$ and an input sentence $\mathbf{x}$. In our work, we adopt word- and sentence-level scorers as in Li et al. (2020).

A word-level scorer focuses on keyword information, where the keywords in the input sentence $\mathbf{x}$ are extracted by the Rake system (Rose et al., 2010). As for each keyword, we find the closest word in the candidate sentence $\mathbf{x}$ in terms of cosine similarity. Then, the pretrained RoBERTa model (Liu et al., 2019) is adopted to compute the contextualized representation, denoted by $\text{RBT}(\text{w}, \mathbf{s})$, for a word w in some sentence $\mathbf{s}$.

The word-level score is defined as the lowest similarity among all the keywords, encouraging the output sentence to contain all the keywords of the input. Specifically, the score is given by

$$f_{\text{word}}(\mathbf{y}, \mathbf{x}) = \min_{\text{k}\in\text{keyword}(\mathbf{x})} \max_{\text{y}_i\in\mathbf{y}} \cos(\text{RBT}(\text{k}, \mathbf{x}), \text{RBT}(\text{y}_i, \mathbf{y})) \tag{3.7}$$

where $\text{keyword}(\mathbf{x})$ is the list of keywords existing in sentence $\mathbf{x}$.

A sentence-level scorer computes the cosine similarity of two sentence vectors as

$$f_{\text{sent}}(\mathbf{y}, \mathbf{x}) = \cos(\boldsymbol{y}, \boldsymbol{x}) = \frac{\boldsymbol{y}^\top \boldsymbol{x}}{||\boldsymbol{y}|| \cdot ||\boldsymbol{x}||} \tag{3.8}$$

where the sentence vectors $\boldsymbol{y}$ and $\boldsymbol{x}$ are also encoded by RoBERTa.

In particular, the sentence vectors are obtained by mean pooling of the word embeddings in the sentence. Given a sentence $\mathbf{s} = (\mathrm{s}_1, \mathrm{s}_2, \cdots, \mathrm{s}_N)$, we obtain the sentence vector $\boldsymbol{s}$ by:

$$\boldsymbol{s} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{RBT}(\mathrm{s}_i, \mathbf{s}) \tag{3.9}$$

Here, Equation 3.9 itself serves as the pooling operation, which averages the word embeddings to obtain a single vector representation for the whole sentence.

Finally, the similarity scorer is computed as the product of word- and sentence-level scores:

$$f_{\text{similarity}}(\mathbf{y}, \mathbf{x}) = f_{\text{word}}(\mathbf{y}, \mathbf{x})^{\beta} \cdot f_{\text{sent}}(\mathbf{y}, \mathbf{x})^{\gamma} \tag{3.10}$$

where β and γ are the weighting hyperparameters.

3.4 Discrete Search Algorithm

We perform attribute-transfer generation by discrete search using local editing operations, such as word insertion, deletion, and replacement, following previous work (Li et al., 2020; Liu et al., 2020b). However, we propose to use steepest-ascent hill climbing (SAHC; Russell and Norvig, 2010) as our discrete search algorithm.

3.4.1 Word Editing

The discrete search algorithm performs one word-level edit operation at a time, which changes the candidate sentence locally. We mainly follow Miao et al. (2019) and provide the detail of each edit operation as follows:

Replacement. In the process of editing a candidate sentence $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, \cdots, \mathrm{y}_N)$, we first choose the word at the position i to be replaced with the mask token

"[MASK]". Then, we adopt RoBERTa based on its property of masked language modeling and predict a candidate token at the chosen position.

Due to efficiency concerns, we select top-k candidate words predicted by RoBERTa. Specifically, RoBERTa encodes the whole candidate sentence $\mathbf{y}$ into a contextualized representation $H \in \mathbb{R}^{n \times d}$ where n is the sequence length and d is the dimension of hidden size. Subsequently, the prediction layer in RoBERTa applies linear transformation on H to generate an output matrix $Z = HW^{\top} \in \mathbb{R}^{n \times d_v}$, where $W \in \mathbb{R}^{d_v \times d}$ is the weight matrix of the layer, and d_v represents the PLM's vocabulary size. In the masked word prediction at the chosen position i, RoBERTa utilizes the softmax function over the ith row of the matrix Z, which is represented as the vector $\boldsymbol{z} = (z_1, z_2, \cdots, z_{d_v}) \in \mathbb{R}^{d_v}$. The softmax computation produces a categorical probability distribution P. For every element in the vector $\boldsymbol{z}$, the probability is computed as

$$P_j = \text{softmax}(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{d_v} e^{z_k}} \tag{3.11}$$

where z_j is the jth element in the vector $\boldsymbol{z}$.

We sort the probability distribution P in descending order and establish a list of top-k candidate words based on their predicted probabilities. In this way, we replace the word at the chosen position with each candidate word from the list, thereby generating k candidate sentences considering all the N positions in terms of replacement.

Insertion. We choose a position i either between two words or at the start of a sentence, and then insert a mask token "[MASK]" into it. Same as the operation of replacement, we still adopt RoBERTa to predict and get a list of top-k candidate words based on the predicted probabilities and have k candidate sentences in terms of insertion.

Deletion. We choose the word at the position i and simply remove the word from the sentence to have one candidate output.

During development, we measured the edit distance between the input sentences

and reference outputs for both the sentiment and formality transfer settings. Our observation is that the average edit distance is 2.9 steps for sentiment transfer and 4.7 steps for formality transfer. Therefore, we set the maximum number of edit steps to 5 to maintain their resemblance. This, unfortunately, renders previous search algorithms—such as simulated annealing (SA; Liu et al., 2020b) and first-choice hill climbing (FCHC; Schumann et al., 2020)—ineffective, as they cannot fully make use of limited search steps. In the following subsection, we will introduce our proposed search algorithm, which is suitable for the scenario of conducting the discrete search in a few edit steps.

3.4.2 Steepest-Ascent Hill Climbing Algorithm

In our work, we propose to use steepest-ascent hill climbing (SAHC; Russell and Norvig, 2010) for discrete search. It greedily finds the best edit for every search step and selects the best candidate before reaching the maximum edit steps. As will be detailed in Section 4.5.2, we conduct an analysis of different search discrete algorithms, and our proposed SAHC algorithm can effectively perform attribute-transfer generation, significantly improving efficiency compared with other search algorithms such as SA and FCHC.

Moreover, we design an additional stopping criterion that the search terminates when the prompted PLM predicts that the attribute has been transferred even if it has not reached the maximum edit steps. This not only improves time efficiency but also encourages non-transfer attribute preservation.

We summarize our approach in Algorithm 1, which performs the following steps:

Step 1. The original sentence $\mathbf{x}$ is taken as input, and this is used to initialize the state with $\mathbf{y}^{(0)} = \mathbf{x}$.

Step 2. The search step t loops from 1 and goes up to the maximum edit step T.

Step 3. At a search step t, the SAHC algorithm enumerates all the editing positions and operations.

Algorithm 1 Prompt-Based Editing

Input: Original sentence $\mathbf{x}$, iterative steps T
$\mathbf{y}^{(0)} = \mathbf{x}$
for $t \in \{1, \cdots, T\}$ **do**
    Enumerate all edit positions and operations
    Obtain the highest-scored candidate $\mathbf{y}'$ by Eqn. (3.3)
    if $f_{\text{transfer}}(\mathbf{y}') > 1$ **then** ▷ *PLM believes attribute transferred*
        return $\mathbf{y}'$
    if $f(\mathbf{y}^{(t-1)}, \mathbf{x}) \geq f(\mathbf{y}', \mathbf{x})$ **then** ▷ *Local optimum found*
        return $\mathbf{y}^{(t-1)}$
    else: $\mathbf{y}^{(t)} = \mathbf{y}'$
return $\mathbf{y}^{(T)}$

Step 4. The highest-scored candidate $\mathbf{y}'$ is obtained according to Equation 3.3.

Step 5. If the score $f_{\text{transfer}}(\mathbf{y}')$ is larger than one, it indicates that the PLM believes the attribute of $\mathbf{y}'$ to be transferred to the target one, thereby SAHC returns $\mathbf{y}'$ as the final output and terminates the search.

Step 6. If the overall score of the current sentence $\mathbf{y}^{(t-1)}$ is equal to or greater than $\mathbf{y}'$ that of the candidate $\mathbf{y}^{(t-1)}$, it indicates that the local optimum has been found. In such a case, SAHC returns $\mathbf{y}'$ as the final output and terminates the search. Otherwise, proceed to Step 7.

Step 7. SAHC takes the candidate $\mathbf{y}'$ as the current sentence and goes back to Step 2 for further discrete search.

Step 8. If SAHC has reached the maximum edit step T without early termination, it ends up returning $\mathbf{y}^{(T)}$ as the final search output.

3.5 Summary

In this chapter, I went through three significant aspects of our proposed approach: the prompt-based classifier, search objective, and search algorithm.

I started by demonstrating the framework of the prompt-based classifier, which is designed to transform the prompt-based generation problem into a classification one. Specifically, the classifier is a frozen pretrained language model, and we use a prompt to query the language model to predict the probabilities of certain words.

The probabilities are then used to compute a score of the attribute to be transferred for the following search objective.

I then focused on the search objective for local search and offered a detailed explanation for each scorer individually. The language fluency scorer, the non-transfer attribute similarity scorer, and the above-mentioned transfer scorer are combined to provide the overall score of a candidate sentence. In our case, we used the reciprocal of perplexity to represent language fluency because lower perplexity equals higher fluency; we further incorporated both word- and sentence-level scores for comprehensively representing the similarity of attributes to be preserved between inputs and model outputs.

Our last aspect was the discrete search algorithm. I explained three editing operations in detail, including replacement, insertion, and deletion. Then, I described the SAHC algorithm designed for editing sentences within a limited number of search steps, instead of using FCHC or SA, based on our observation of the attribute transfer tasks. In addition, we designed a stopping criterion to terminate the local search when the candidate sentence's attribute is predicted to be transferred, which largely improves time efficiency and non-transfer attribute preservation.

Chapter 4

Experiments

4.1 Overview

In this chapter, I will first introduce our experimental setups in Section 4.2, including the introduction of four benchmark datasets, implementation details, and different baseline prompting approaches. In Section 4.3, I will provide detailed explanations for different evaluation metrics. Section 4.4 will focus on the main results on four datasets. Then, I will show a series of detailed empirical analyses in Section 4.5, including ablation study, analysis of different discrete search algorithms, delimiter pairs, and editing operations. Finally, I will provide case studies of model outputs in Section 4.6.

4.2 Experimental Setup

4.2.1 Datasets

We evaluated our prompt-based editing approach on three attribute transfer tasks: sentiment, formality, and Shakespeare-to-modern transfer. We divide these tasks into two settings:

- **No meaning-preserving:** Sentiment transfer does not preserve the meaning during the transfer process.
- **Meaning-preserving:** Formality and Shakespeare-to-modern transfer pre-

serve the meaning during the transfer process.

Both of the settings are addressed by our prompt-based editing approach.

We used Yelp reviews (YELP; Zhang et al., 2015) and Amazon reviews (AMAZON; He and McAuley, 2016) for sentiment transfer. These two datasets have been widely used in previous work (Li et al., 2018; Luo et al., 2019; John et al., 2019; Reif et al., 2022; Suzgun et al., 2022). YELP contains reviews for restaurants and other businesses, while AMAZON contains product reviews that were obtained from the Amazon website. Both YELP and AMAZON datasets contain 500 positive and 500 negative sentences in the test set.

Then, we used Grammarly's Yahoo Answers Formality Corpus (GYAFC; Rao and Tetreault, 2018) for formality transfer. This dataset is also widely used in previous work (Luo et al., 2019; Lai et al., 2021; Reif et al., 2022; Suzgun et al., 2022). GYAFC consists of sentences that were extracted from a question-answering forum (Yahoo Answers). We chose the "Family & Relationships" domain, following Luo et al. (2019) and Suzgun et al. (2022). The test set contains 500 formal and 500 informal sentences.

In addition, we adopted the SHAKESPEARE dataset (Xu et al., 2012) for Shakespeare-to-modern transfer and used the test set provided in Suzgun et al. (2022) for a fair comparison. The test set contains 599 Shakespeare sentences from William Shakespeare's *Romeo and Juliet*, written in Shakespeare and modern English.

An overview of the datasets is shown in Table 4.1.

4.2.2 Implementation Details

We utilized Eleuther AI's off-the-shelf GPT-J-6B as the prompt-based classifier. GPT-J-6B features 28 encoder layers, each having 16 attention heads. Further, we used a non-finetuned pretrained language model RoBERTa-Large (Liu et al., 2019) to encode the sentences, and to predict top-k words as candidate edits (Section 3.3). We set $k = 50$ for all four datasets.

Dataset	Attribute	Example Sentence Pair	Size
Yelp	Negative	moving past the shape, they were dry and truly tasteless.	500
	Positive	moving past the shape, they were dry and truly tasty.	500
Amazon	Negative	nokia knows how to design a very terrible interface.	500
	Positive	nokia knows how to design a very good interface.	500
Gyafc	Informal	and so what if it's a rebound relationship for both of u?	500
	Formal	what if it is a rebound relationship for both of you?	500
Shakespeare	Shakespeare	is rosaline, whom thou didst love so dear, so soon forsaken?	599
	Modern	have you given up so quickly on rosaline, whom you loved so much?	599

Table 4.1: Overview of text attribute transfer datasets used in this work

Regarding the weighting hyperparameters[1] α, β, and γ of the search objective $f(\mathrm{y})$ in Eqn. (3.3), they are $\frac{1}{4}$, $\frac{1}{6}$, and $\frac{1}{6}$ for both Yelp and Amazon datasets, and $\frac{1}{4}$, $\frac{3}{8}$, and $\frac{3}{8}$ for both Gyafc and Shakespeare datasets.

Our proposed approach was developed with Python 3.7 and Pytorch 1.11.0. The experiments were conducted on NVIDIA A100 SXM4 GPUs.

4.2.3 Baseline Approaches

We compared our proposed method with one naive and three competing baseline approaches:

- **Add "not".** This method adds the word "not" at the start of a sentence, which is specifically applied to sentiment transfer.

- **Vanilla Prompting.** This baseline method queries a PLM with the prompt

  Here is some text: {[**x**]}. Here is a rewrite of the text, which is more [s]: {

  where [**x**] is the input and [s] is the attribute to be transferred (e.g., *positive* and *negative* in sentiment transfer, and *formal* and *informal* in formality transfer) to directly obtain an attribute-transferred sentence, as shown in Figure 2.3. In

[1]We used the hyperparameters in Li et al. (2020) and did not tune them.

Dataset	[Few-Shot Examples] **and** [Test-Time Input]
YELP	The sentiment of the text {this place is awful!} is: positive \n ### \n The sentiment of the text {this place is amazing!} is: positive \n ### \n The sentiment of the text {I hated their black tea and hated hot chocolate selections!} is: negative \n ### \n The sentiment of the text {I hated their black tea and hated hot chocolate selections!} is: positive \n ### \n The sentiment of the text{it's small yet they make you feel right at home.} is:
GYAFC	The formality of the text {ohhh i dont́ intend to be mean ...} is: informal \n ### \n The formality of the text {i do not intend to be mean} is: formal \n ### \n The formality of the text {,,, that sucks man but u gotta move on :)} is: formal \n ### \n The formality of the text {that is unfortunate, but you need to move on} is: formal \n ### \n The formality of the text {and so what if it is a rebound relationship for both of you ?} is:
SHAKESPEARE	The genre of the text {what hast thou there ?} is: old \n ### \n The genre of the text {i do not intend to be mean} is: modern \n ### \n The genre of the text {talk not to me, for i'll not speak a word.} is: formal \n ### \n The genre of the text {don't talk to me, because i won't answer you.} is: modern \n ### \n The genre of the text {as mine on hers, so hers is set on mine, and all combined, save what thou must combine by holy marriage.} is:

Table 4.2: A complete list of exemplars used in our few-shot experiments for all attribute transfer tasks. Here, the color gray is used to highlight the examples used in the few-shot prompt and the color violet represents a test-time input example.

all attribute transfer tasks, we used four exemplars in the few-shot setting, as displayed in Table 4.2.

- **Distant-Exemplar Prompting.** We adopted the approach in Reif et al. (2022), which queries a large-scale PLM (such as LLM, LLM-dialog, and 175-billion-parameter GPT-3) with several attribute-transferred exemplars in a few-shot manner. However, their exemplars have a different label from the test cases, and thus we call it *distant-exemplar prompting*. We use the same prompt provided by Reif et al. (2022) to obtain results with the GPT-3 curie (6.7B) and the off-the-shelf GPT-J-6B for the four benchmark datasets.

- **Prompt & Rerank.** This prompting method (Suzgun et al., 2022) proposes to generate multiple candidate outputs from different manually designed prompts (e.g., different textual templates or different delimiter pairs); then, they rerank the outputs by a heuristically defined scoring function. It should be mentioned that the paper (Suzgun et al., 2022) adopts a setting that is non-compatible with prior work; specifically, they report different directions of sentiment transfer separately, while excluding informal-to-formal transfer in the formality experiment. Therefore, we replicated their work under the standard settings (Luo et al., 2019; Reif et al., 2022).

To the best of our knowledge, distant-exemplar prompting (Reif et al., 2022) and Prompt & Rerank (Suzgun et al., 2022) are the only prior studies that conduct prompting methods on text attribute transfer. In the following subsection, I will introduce the metrics we used for automatic evaluation.

4.3 Evaluation Metrics

We evaluated different approaches in terms of the following evaluation metrics.

Attribute transfer accuracy. The accuracy score measures the success rate of attribute-transferred model outputs. Following the practice in Reif et al. (2022), Lai et al. (2021), and Krishna et al. (2020), we used several pretrained classifiers for attribute classification to determine whether a generated output possesses the desired attribute. In particular, SiEBERT (Hartmann et al., 2022) is used for sentiment classification, an off-the-shelf RoBERTa-Large (Liu et al., 2019) is fine-tuned separately for formality classification and Shakespeare-to-modern classification.

Given a corpus of model outputs $\mathcal{D} = \{\mathbf{y}^{(m)}, r^{(m)}\}_{m=1}^{M}$, where the target label $r \in \{0, 1\}$[2] and M is the number of model outputs in one attribute transfer task, the accuracy score (ACC) is then computed as the success rate of attribute-transferred

[2]In sentiment transfer, we denote *negative* as 0 and *positive* as 1; in formality transfer, we denote *informal* as 0 and *formal* as 1; in Shakespeare-to-modern transfer, we denote *old* as 0, *modern* as 1.

sentences:

$$\text{ACC} = \frac{\sum_{m=1}^{M} \mathbf{1}\{\text{C}(\mathbf{y}^{(m)}) = r^{(m)}\}}{M} \tag{4.1}$$

where $\text{C}(\cdot)$ is the function of binary classification of a sentence, and $\mathbf{1}(\cdot)$ is the indicator function that it is 1 when the condition is true or 0 when false.

BLEU. The BLEU score measures the similarity between the model outputs and human-written reference sentences that are provided by (Luo et al., 2019). It is commonly used in research of text attribute transfer (Shen et al., 2017; Dai et al., 2019; Lyu et al., 2021) to measure the preservation of non-transfer attributes. In this work, we used the script `multi-bleu.perl` to obtain the BLEU-4 score following Luo et al. (2019) and Reif et al. (2022).

The script calculates the n-gram precision of a candidate sentence with respect to a list of reference sentences. Precision is cut by the maximum occurrence of each n gram across different references, preventing the system from repeating common words. To evaluate the generated texts at the corpus level, the n-gram precision p_n is computed as

$$p_n = \frac{\sum_{C \in \text{Cands}} \sum_{n\text{-gram} \in C} \text{Count}_{\text{clip}}(n\text{-gram})}{\sum_{C \in \text{Cands}} \sum_{n\text{-gram} \in C} \text{Count}(n\text{-gram})} \tag{4.2}$$

where "Cands" represents the candidate sentences, and the $\text{Count}_{\text{clip}}(\cdot)$ operation counts the number of matching n-grams clipped to the maximum occurrences in reference sentences, while the $\text{Count}(\cdot)$ operation counts the number of n-grams in the candidate sentences (Wen et al., 2022).

Second, the brevity penalty (BP) is applied to precision p_n to penalize overly short generations and ensure they cannot gain high performance:

$$\text{BP}(r, c) = \begin{cases} e^{1-\frac{r}{c}} & \text{if } r \leq c \\ 1 & \text{if } r > c \end{cases} \tag{4.3}$$

where c is the sum of generated sentences' length in the corpus, and r is the sum of reference sentences' length.

Finally, the BLEU score is computed as the weighted geometric mean of n-gram

precisions for $n = 1, \cdots, 4$:

$$\text{BLEU} = \text{BP} \cdot \exp\big(\frac{1}{4}\sum_{n=1}^{4} \log p_n\big) \tag{4.4}$$

BERTScore. It is a powerful language generation evaluation metric based on pretrained BERT contextual embeddings (Zhang et al., 2020a). This metric computes the semantic similarity of a pair of sentences as a sum of cosine similarities between the embeddings of their tokens. BERTScore (BS) can capture deep semantic meanings and contextual nuances of the texts and has been widely used in previous work (Liu and Liu, 2021; Li and Liang, 2021; Liu et al., 2022b).

Specifically, BERTScore matches each token in the input sentence $\mathbf{x}$ to a token in the reference sentence $\mathbf{y}$ to compute recall, and each token in $\mathbf{y}$ to a token in $\mathbf{x}$ to compute precision. Greedy matching is used to maximize the similarity score, where each token is matched to the most similar token in the other sentence. Further, precision and recall are combined to compute an F1 score. For a reference sentence $\mathbf{y}$ and candidate $\mathbf{x}$, the recall, precision scores are

$$R_{\text{BERT}} = \frac{1}{|\mathbf{y}|}\sum_{\text{y}_i \in \mathbf{y}} \max_{\text{x}_j \in \mathbf{x}} \boldsymbol{y}_i^\top \boldsymbol{x}_j, \quad P_{\text{BERT}} = \frac{1}{|\mathbf{x}|}\sum_{\text{x}_j \in \mathbf{x}} \max_{\text{y}_j \in \mathbf{y}} \boldsymbol{y}_i^\top \boldsymbol{x}_j \tag{4.5}$$

where $\boldsymbol{x}_j$ and $\boldsymbol{y}_i$ are token embeddings. Thus, the F1 score (BERTScore) is:

$$\text{BERTScore} = F_{\text{BERT}} = 2 \cdot \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \tag{4.6}$$

Geometric and Harmonic Mean. We follow Luo et al. (2019) and adopt the geometric mean (GM) and harmonic mean (HM) to evaluate the overall performance based on the transfer accuracy mentioned above and the BLEU score. Given N attribute-transferred sentences, we first calculate the accuracy (ACC) score and BLEU score based on Equations 4.1 and 4.4, respectively, and then calculate the GM score in terms of accuracy and BLEU score as follows:

$$\text{GM} = \sqrt[2]{\text{ACC} \cdot \text{BLEU}} \tag{4.7}$$

Setting	Method	Model	#Para	Yelp					Amazon				
			(B)	ACC	BLEU	BS	GM	HM	ACC	BLEU	BS	GM	HM
-	Add "not"	-	0	29.2	58.4	94.7	41.3	38.9	36.6	43.6	93.2	39.9	39.8
zero-shot	Vanilla	GPT-J-6B	6	63.7	34.5	90.4	46.9	44.8	56.9	26.6	88.8	38.9	36.3
		LLM	128	69.7*	28.6*	-	44.6	40.6	-	-	-	-	-
		LLM-dialog	128	59.1*	17.6*	-	32.3	27.1	-	-	-	-	-
	P&R	GPT-J-6B	6	68.6	19.8	83.2	35.2	30.1	57.1	21.7	86.6	35.2	31.4
	Ours	GPT-J-6B	6	**73.0**	**40.1**	**92.7**	**54.1**	**51.7**	**72.7**	**28.6**	**91.1**	**45.6**	**41.0**
few-shot	Distant exemplars	GPT-J-6B	6	52.8	35.8	-	43.5	42.7	51.0	27.1	90.9	37.2	35.4
		GPT-3 curie	6.7	53.0*	48.3*	-	50.6	50.5	72.2	22.9	87.5	40.7	34.8
		LLM	128	79.6*	16.1*	-	35.8	26.8	-	-	-	-	-
		LLM-dialog	128	**90.6***	10.4*	-	30.7	18.7	-	-	-	-	-
		GPT-3 danvinci	175	74.1*	43.8*	-	57.0	55.1	**87.3**	28.3	91.0	49.7	42.7
	P&R	GPT-J-6B	6	75.0	42.5	94.0	56.5	54.3	66.8	20.5	86.0	37.0	31.4
	Ours	GPT-J-6B	6	74.5	**48.9**	**94.4**	**60.3**	**59.0**	78.5	**37.1**	**92.5**	**54.0**	**50.4**

Table 4.3: Results on Yelp and Amazon test sets. #Para: Number of parameters. GM and HM: Geometric mean and harmonic mean of ACC and BLEU. BS: BERTScore F1-score scaled between 1 to 100. †We replicated Prompt & Rerank (Suzgun et al., 2022) by their released code, as the settings in Suzgun et al. (2022) are incompatible with other previous work. Results with * are quoted from (Reif et al., 2022). Other results are given by our experiments. The performance of LLM and LLM-dialog is not available for Amazon because these PLMs are not public.

Method	Model	#Para (B)	ACC	BLEU	BS	GM	HM
Distant exemplars	GPT-J-6B	6	39.4	33.1	91.7	36.1	36.0
P&R	GPT-J-6B	6	**44.4**	32.9	91.4	38.2	37.8
Ours	GPT-J-6B	6	**44.4**	**33.4**	**92.7**	**38.5**	**38.1**

Table 4.4: Four-shot performance on the Gyafc dataset, considering both directions of *informal* to *formal* and *formal* to *informal*.

Similarly, we calculate the HM score as

$$\text{HM} = \frac{2}{\frac{1}{\text{ACC}} + \frac{1}{\text{BLEU}}} \tag{4.8}$$

Again, this calculation follows the standard practice in previous work (Luo et al., 2019; Li et al., 2020).

Method	Model	#Para (B)	ACC	BLEU	BS	GM	HM
Distant exemplars	GPT-J-6B	6	61.0	19.1	87.1	34.1	29.1
P&R	GPT-J-6B	6	**70.2**	21.0	88.0	**38.4**	**32.3**
Ours	GPT-J-6B	6	57.7	**21.5**	**88.3**	35.2	31.3

Table 4.5: Four-shot performance on the SHAKESPEARE dataset, considering the same Shakespeare-to-modern direction as in Suzgun et al. (2022).

4.4 Experimental Results

Table 4.3 shows the performance of different prompting systems on the YELP and AMAZON datasets. We observe that our approach achieves the best geometric mean (GM) and harmonic mean (HM) scores among all the approaches. Compared with the recently proposed prompting system, Prompt & Rerank (Suzgun et al., 2022), our approach outperforms by more than 14 and 3 points for GM, and 15 and 5 points for HM in the zero- and few-shot settings, respectively, averaged across the two datasets. Further, compared with 175-billion-parameter GPT-3 with distant exemplars (i.e., attribute-transferred exemplars containing source texts and outputs written in non-target labels), our approach yields higher GM and HM scores by more than 3, and 5 points, respectively, also averaged across the two datasets. This is a compelling result, as our approach yields a better balance between non-transfer attribute preservation and attribute transfer strength while using a 20x smaller PLM.

Table 4.4 shows the results of different prompting systems on the GYAFC dataset, where both informal-to-formal and formal-to-informal directions are considered (Luo et al., 2019; Reif et al., 2022). To ensure a fair comparison with previous prompting systems, we followed Suzgun et al. (2022) and conducted experiments in a four-shot setting. As shown, our approach outperforms previous approaches in GM and HM scores, which is consistent with the results in Table 4.3. It is also noticed that our approach achieves less improvement on GYAFC than on YELP and AMAZON, as

formality transfer is more challenging than sentiment transfer.

In addition, Table 4.5 shows the results of different prompting systems on the SHAKESPEARE dataset. We considered the Shakespeare-to-modern direction and conducted experiments in a four-shot setting following Suzgun et al. (2022). As shown, our approach does not achieve the best performance in GM and HM scores compared to P&R. One possible reason is that the SHAKESPEARE dataset shows more word variations between input sentences and references. It also demands more word reordering, making it more challenging than sentiment and formality transfer.

4.5 Detailed Analyses

In this section, we conduct in-depth analyses to assess the effectiveness of our prompt-based editing approach. Due to limited time and resources, we chose the sentiment transfer datasets, YELP and AMAZON, as our testbed.

4.5.1 Ablation Study

To evaluate the contribution of key components in our model, we conducted an ablation study of different scoring functions and our proposed stopping criterion.

Table 4.6 shows that all the scorers play a role in our approach, and that the prompt-based scorer f_{transfer} is the most important one. This makes sense, as it is the only signal of the attribute to be transferred. Without the scorer f_{transfer}, we would not be able to perform meaningful attribute transfer. Moreover, we find that the fluency scorer f_{fluency} slightly hurts attribute transfer accuracy and BLEU scores, which are the standard metrics in Luo et al. (2019). However, it significantly hurts language fluency, characterized by an increase in perplexity (PPL) that roughly estimates the fluency of text (John et al., 2019). Therefore, we deem the fluency scorer f_{fluency} essential to our model.

In addition, our approach involves a stopping criterion that is designed to terminate the search process if the PLM believes the certain attribute is successfully transferred.

Dataset	Model	ACC	BLEU	BS	GM	HM	PPL
YELP	Full model	73.0	**40.1**	**92.7**	54.1	51.7	122.7
	w/o f_{transfer}	17.9	25.1	84.2	21.2	33.9	29.3
	w/o $f_{\text{similarity}}$	74.0	39.0	92.0	53.7	51.1	124.0
	w/o f_{fluency}	**81.3**	39.3	92.3	**56.5**	**53.0**	223.6
	w/o stop criterion	78.3	25.2	81.2	44.4	38.1	192.4
AMAZON	Full model	72.7	**28.6**	**91.1**	45.6	41.0	137.2
	w/o f_{transfer}	33.6	20.2	85.7	26.1	25.3	31.5
	w/o $f_{\text{similarity}}$	71.1	28.1	90.9	44.7	40.3	116.3
	w/o f_{fluency}	78.0	**28.6**	**91.1**	**47.2**	**41.8**	229.9
	w/o stop criterion	**79.9**	19.3	84.9	39.3	31.1	176.3

Table 4.6: Ablation study on the sentiment transfer datasets in the zero-shot setting. PPL: Perplexity (the smaller, the better). In the "w/o f_{transfer}" setting, the model mainly optimizes toward f_{fluency}, so it achieves an extraordinarily low PPL; however, its designated attribute is usually not transferred, shown by extraordinarily low ACC. Therefore, this is not a meaningful attribute transfer setting.

As seen from the last row of Table 4.6, more edit steps (w/o stop criterion) improve the attribute transfer accuracy but drastically hurt BLEU and BERTScore. This shows that our stopping criterion is able to seek a balance between the transfer strength and preserving the non-transfer attribute of original texts.

4.5.2 Analysis of Discrete Search Algorithms

Our steepest-ascent hill climbing (SAHC) algorithm enumerates candidate edits, including word deletion, insertion, and replacement (where top-50 candidate words are considered for efficiency concerns). Then, SAHC selects the best one for the next round of editing, shown in Algorithm 1.

We compared our SAHC with two stochastic optimization algorithms, first-choice hill climbing (FCHC; Schumann et al., 2020) and simulated annealing (SA; Liu et al., 2020b), detailed as follows.

Dataset	Algorithm	ACC	BLEU	BS	GM	HM
Yelp	SAHC	**73.0**	**40.1**	**92.7**	**54.1**	**51.7**
	FCHC	67.2	31.8	89.7	46.2	43.1
	SA	66.0	28.7	89.1	43.5	40.0
Amazon	SAHC	**72.7**	**28.6**	**91.1**	**45.6**	**41.0**
	FCHC	64.1	24.8	90.1	39.8	35.7
	SA	63.2	23.7	88.9	38.7	34.4

Table 4.7: Results of different discrete search algorithms on Yelp and Amazon datasets.

- **FCHC.** During the search process, FCHC iteratively applies stochastic local changes to an input sentence. If the candidate sentence is better than the current one, the algorithms will accept the proposed one. Otherwise, it rejects the candidate and retains the current one.

- **SA.** SA also applies iterative stochastic word changes to an input sentence, but its acceptance criterion is different from FCHC. SA can accept a better candidate, and it may additionally accept the proposed candidate with a small probability, even if it is worse than the current sentence.

Table 4.7 shows that our SAHC algorithm significantly outperforms FCHC and SA in both attribute transfer accuracy and the BLEU score, indicating that SAHC in our scenario is more suited than other discrete search algorithms. This is likely due to the limited number of edit steps, which requires that the algorithm should make an effective edit at every search step.

4.5.3 Delimiter Pairs

In practice, we have not performed intensive prompt engineering but simply adopted the most intuitive expression, as shown in Equation 3.1. However, it is known that the delimiter pairs in the prompt may affect the model performance (Reif et al., 2022;

Suzgun et al., 2022). Therefore, we conducted a detailed analysis of different delimiter pairs' effects on the quality of model outputs.

Following Suzgun et al. (2022), we conducted experiments with ten delimiter pairs. Specifically, these ten pairs are: (1) curly brackets $\{\cdot\}$, (2) square brackets $[\cdot]$, (3) angle brackets $\langle\cdot\rangle$, (4) parentheses $(\cdot)$, (5) quotes " $\cdot$ ", (6) dashes $-\cdot-$, (7) triple angle brackets $\langle\langle\langle\cdot\rangle\rangle\rangle$, (8) bracket quotes $\rangle$ " $\cdot$ ", (9) asterisk quotes * " $\cdot$ ", and (10) double curly brackets $\{\{\cdot\}\}$.

We refactored the prompt in Equation 3.1 with each of the above-mentioned delimiter pairs. For example, if we have an input sentence, " *I like playing tennis!* ", for sentiment transfer, then the prompt using square brackets would be

The sentiment of the text [*I like playing tennis !*] is:

Table 4.8 reports the results. We find the delimiter pair "$\langle\cdot\rangle$" yields the best overall results, achieving the best geometric mean (GM) and harmonic mean (HM) scores on both datasets, with a GM of 59.4 and an HM of 57.7 on Yelp as well as a GM of 49.7 and an HM of 45.0 on Amazon. On the contrary, the delimiter pair "$(\cdot)$" yields the lowest performance in terms of all evaluation metrics.

Nevertheless, in our main experiments, we used the curly brackets following Reif et al. (2022) and Suzgun et al. (2022) for a fair comparison. Our analysis here indicates the potential of our approach, suggesting that we can further improve model performance through some prompt engineering techniques.

4.5.4 Editing Operations

To evaluate the contribution of each edit operation (word deletion, replacement, and deletion), we conducted experiments on sentiment and formality transfer datasets. Specifically, we computed the proportion of each operation used in the search process for each dataset.

The result is shown in Table 4.9. We observe that word replacement is predominant in the search process in both Yelp and Amazon datasets, likely because of replacing

Delimiter Pairs	Yelp					Amazon				
	ACC	BLEU	BS	GM	HM	ACC	BLEU	BS	GM	HM
{·}	73.0	40.1	92.7	54.1	51.7	72.7	28.6	91.1	45.6	41.0
[·]	68.7	39.9	92.5	52.4	50.5	74.0	28.5	91.1	45.9	41.1
⟨·⟩	76.0	**46.5**	**94.0**	**59.4**	**57.7**	**78.5**	31.5	92.2	**49.7**	**45.0**
(·)	62.8	37.8	92.0	51.0	48.8	66.7	27.7	90.6	43.0	39.1
" · "	70.0	45.1	93.6	56.2	54.8	70.2	30.5	91.7	46.3	42.5
– · –	68.9	40.3	92.8	52.7	50.9	71.1	29.3	91.3	45.6	41.5
⟨⟨⟨·⟩⟩⟩	**77.0**	43.9	93.0	58.1	55.9	74.1	29.5	91.3	46.8	42.2
⟩ " · "	73.0	42.7	92.9	55.8	53.8	70.4	**32.7**	**92.8**	48.0	44.7
* " · "	68.7	41.2	92.8	53.2	51.5	71.2	29.9	91.5	46.1	42.1
{{·}}	73.5	44.7	93.4	57.3	55.6	70.8	30.1	91.6	46.2	42.2

Table 4.8: Results on Yelp and Amazon datasets with ten types of delimiter pairs.

Dataset	Edit Operation		
	Replacement	Insertion	Deletion
Yelp	**53.4%**	37.8%	8.8%
Amazon	**52.2%**	39.1%	8.7%
Gyafc	39.9%	**47.5%**	12.6%

Table 4.9: Proportion of all three editing operations in Yelp, Amazon and Gyafc datasets.

adjectives or verbs to convey a different sentiment. On the other hand, word insertion contributes the most in the Gyafc dataset, indicating that formality transfer may require adding new words to make sentences more formal or informal. Overall, all three operations play a role in text attribute transfer, and their contributions vary across datasets.

4.6 Case Study

We show in Table 4.10 that our method is able to mitigate the error accumulation issue in an autoregressive generation. This is observed through several example outputs by

P&R and our approach for YELP, AMAZON, and GYAFC datasets. We saw that the previous method, which performs autoregressive generation, generates unsatisfactory and less controllable sentences. For example, given the source input *for my purpose this is the perfect item* in the positive-to-negative sentiment transfer of the AMAZON dataset, P&R generates an unrelated sentence starting with *So this text has*, which leads to the subsequent improper word predictions *a text and to be a rewrite*.

By contrast, our prompt-based editing approach transfers the sentiment of a source sentence from positive to negative by inserting the words *but* and *not*, while maintaining other words from input sentences. This shows that our approach is able to generate more sensible and controllable sentences.

In addition, we discover that our approach is able to convert certain attributes of source inputs with multiple edits. For example, given the source sentence *i'm unsure concerning what i should do* in formal-to-informal transfer, our approach inserts multiple tokens (*yeah*, *lol*, *really*, and "...") at the beginning and replaces *should* with *'ll* at the end. In this way, the sentence is transferred to an informal one. By allowing iterative edits and examining all possible positions and editing operations, we enable multiple word-level edits scattered throughout the sentence and experience a gradual attribute transfer.

4.7 Summary

In this chapter, we started by introducing the details of two sentiment transfer datasets, YELP and AMAZON, one formality transfer dataset, GYAFC, and one Shakespeare-to-modern transfer dataset, SHAKESPEARE. We further evaluated attribute transfer strength, non-transfer attribute preservation, and overall performance of our approach in comparison with baseline methods. We used pretrained classifiers to calculate the classification accuracy, thereby measuring attribute transfer strength; we adopted the widely used BLEU and BERTScore to measure semantic similarity between model outputs and reference sentences. In addition, we utilized both geometric mean and

Yelp	**Negative ⟶ Positive**	**Positive ⟶ Negative**
Source	so far i'm not really impressed	their lunch special is a great value
P&R	*The text is good now*	but their lunch is a *great* value
Ours	so far i'm really impressed	their lunch special is not a great value
Amazon	**Negative ⟶ Positive**	**Positive ⟶ Negative**
Source	i like neutrogena products as a rule, so this was a disappointment.	for my purpose this is the perfect item.
P&R	i like neutrogena products, so this was a *disappointment.*	for my purpose this is the *perfect* item. *So this text has two different purposes: to be a text and to be a rewrite...*
Ours	overall i like neutrogena products as a rule, so this was a success.	but for my purpose this is not the perfect item.
Gyafc	**Informal ⟶ Formal**	**Formal ⟶ Informal**
Source	think about what good it brought about.	i'm unsure concerning what i should do.
P&R	think about what good it *will bring about ...*	i'm *not certain* about what to do *next...*
Ours	please think about what all the good news has brought about.	yeah lol really ... i'm unsure concerning what i 'll do.

Table 4.10: Example outputs on the YELP, AMAZON, and GYAFC datasets. Double directions (positive ↔ negative and formal ↔ informal) are shown, and improperly generated words are italicized.

harmonic mean between accuracy and the BLEU score to measure the overall performance of our approach.

Finally, we conducted experiments on those multiple benchmark datasets and demonstrated that our prompt-based editing approach outperforms existing systems on most datasets. In addition, we further conducted a series of detailed analyses of the generated outputs. These analyses suggest that our scoring functions and discrete search algorithm can generate controllable attribute-transferred sentences. Overall, the results indicate that our prompt-based editing approach can alleviate the error accumulation issue that exists in autoregressive generation and also achieve a strong balance between transfer strength and non-transfer attribute preservation.

Chapter 5

Conclusion

5.1 Thesis Summary

Prompting with large language models has shown its capacity for autoregressive generation in a zero-shot or few-shot manner on different kinds of text generation tasks. However, such autoregressive generation is less controllable, as words are generated one after another by the pretrained language model (PLM), and early prediction errors of the PLM will affect its future word predictions, leading to low performance in general.

In this thesis, we proposed a novel prompt-based editing approach to text attribute transfer, which transforms a prompt-based generation problem into a classification one. The approach consists of three main contributions: prompt-based classifier, search objective, and discrete search algorithm. First, we introduced the framework of the prompt-based classifier, which is designed to predict the probability of a certain attribute value, functioning as the score of the attribute to be transferred. Second, we introduced language fluency and non-transfer attribute similarity scorers, which are combined with the transfer scorer to control the quality of generated model outputs. Finally, we proposed to use the steepest-ascent hill climbing (SAHC) algorithm to search for the best candidate sentence in each search step. We further designed a stopping criterion to terminate local search when the candidate sentence's attribute is transferred. Consequently, SAHC can perform attribute transfer within a small

number of edit steps, which largely improves time efficiency and helps preserve the non-transfer attributes.

We conducted an automatic evaluation on four attribute transfer datasets, and the experimental results show that our approach largely outperforms the existing prompting systems on multiple datasets. Moreover, we conduct quantitative analyses to demonstrate the effectiveness of our scoring functions and discrete search algorithm, and qualitative analyses to verify that our approach highlights the balance between attribute transfer strength and preservation of other attributes. Overall, our experiments show that our prompt-based editing approach is able to mitigate the error accumulation issue in the autoregressive generation process.

5.2 Limitations and Future Work

One limitation of our approach is that local edits cannot help rewrite sentences to a large extent. This restricts our method to relatively simple attribute transfer tasks. Inspired by Kumar et al. (2020), we aim to design reordering algorithms that help the model tackle more complex attribute transfer tasks, such as Shakespeare-to-modern transfer.

Another limitation of our approach is the inference efficiency, which may not suffice for real-life applications. A potential solution to our algorithm is to implement it in a highly parallel manner when evaluating different candidates, within merely five iterations. Therefore, the efficiency of our SAHC can be much higher than other search algorithms (such as SA), which requires several hundred search steps (Liu et al., 2020b). Further, the efficiency can also be improved by learning from the search results (Li et al., 2020), i.e., fine-tuning a PLM based on our outputs.

In addition, we apply our approach to only three English-based attribute transfer tasks. In the future, we aim to extend our approach to other languages as well as other attribute transfer tasks.

Further, it is also important to alleviate the need for manually designed prompts.

Currently, our work adopts the most straightforward and intuitive prompt design, without incorporating extensive prompt engineering techniques. In the future, we aim to investigate prompt tuning (Schick and Schütze, 2021b; Li and Liang, 2021; Wei et al., 2022a) as a means to mitigate the reliance on designing prompts.

Bibliography

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

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 Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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

Hiroyuki Deguchi, Taro Watanabe, Yusuke Matsui, Masao Utiyama, Hideki Tanaka, and Eiichiro Sumita. 2023. Subset retrieval nearest neighbor machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 174–189.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402.

Yao Fu, Hao Zhou, Jiaze Chen, and Lei Li. 2019. Rethinking text attribute transfer: A lexical analysis. In *Proceedings of the International Conference on Natural Language Generation*, pages 24–33.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 3816–3830.

Navita Goyal, Balaji Vasan Srinivasan, N Anandhavelu, and Abhilasha Sancheti. 2021. Multi-style transfer with discriminative feedback on disjoint corpus. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3500–3510.

Jochen Hartmann, Mark Heitmann, Christian Siebert, and Christina Schamp. 2022. More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40(1):75–87.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

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 International Conference on World Wide Web*, pages 507–517.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. 2020. Hooks in the headline: Learning to generate headlines with controlled styles. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 5082–5093.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

Shailza Jolly, Zi Xuan Zhang, Andreas Dengel, and Lili Mou. 2022. Search and learn: Improving semantic coverage for data-to-text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10858–10866.

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. *arXiv preprint arXiv:2001.08361*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, pages 22199–22213.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 737–762.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021. Thank you BART! Rewarding pre-trained models improves formality style transfer. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 484–494.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018b. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

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 Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Jingjing Li, Zichao Li, Lili Mou, Xin Jiang, Michael Lyu, and Irwin King. 2020. Unsupervised text generation by learning from search. In *Advances in Neural Information Processing Systems*, pages 10820–10831.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1865–1874.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 4582–4597.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 5315–5333.

Dayiheng Liu, Yeyun Gong, Yu Yan, Jie Fu, Bo Shao, Daxin Jiang, Jiancheng Lv, and Nan Duan. 2020a. Diverse, controllable, and keyphrase-aware: A corpus and method for news multi-headline generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 6241–6250.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Puyuan Liu, Chenyang Huang, and Lili Mou. 2022a. Learning non-autoregressive models from search for unsupervised sentence summarization. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 7916–7929.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020b. Unsupervised paraphrasing by simulated annealing. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 302–312.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 1065–1072.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022b. BRIO: Bringing order to abstractive summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 2890–2903.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 5116–5122.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2116–2138.

Xinyao Ma, Maarten Sap, Hannah Rashkin, and Yejin Choi. 2020. PowerTransformer: Unsupervised controllable revision for biased language correction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 7426–7441.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. CGMH: Constrained sentence generation by Metropolis-Hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6834–6842.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 5316–5330.

Lili Mou. 2022. Search and learning for unsupervised text generation. *AI Magazine*, 43(4):344–352.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 189–194.

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training Gopher. *arXiv preprint arXiv:2112.11446*.

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. *Journal of Machine Learning Research*, 21(140):1–67.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 129–140.

Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, pages 3932–3944.

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 Annual Meeting of the Association for Computational Linguistics*, pages 837–848.

Parker Riley, Noah Constant, Mandy Guo, Girish Kumar, David C Uthus, and Zarana Parekh. 2021. TextSETTR: Few-shot text style extraction and tunable targeted restyling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 3786–3800.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, pages 1–20.

Stuart J Russell and Peter Norvig. 2010. *Artificial Intelligence: A Modern Approach*. Pearson Education Limited.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Timo Schick and Hinrich Schütze. 2021a. Few-shot text generation with natural language instructions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 390–402.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova, and Katja Markert. 2020. Discrete optimization for unsupervised sentence summarization with word-level extraction. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 5032–5042.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex

Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

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 Conference on Empirical Methods in Natural Language Processing*, pages 2195–2222.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. LaMDA: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

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 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2020. Formality style transfer with shared latent space. In *Proceedings of the International Conference on Computational Linguistics*, pages 2236–2249.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022c. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, pages 24824–24837.

Yuqiao Wen, Guoqing Luo, and Lili Mou. 2022. An empirical study on the overlapping problem of open-domain dialogue datasets. In *Proceedings of the Language Resources and Evaluation Conference*, pages 146–153.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 409–420.

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 1980–1984.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 7361–7373.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of the International Conference on Computational Linguistics*, pages 2899–2914.

Min Yang, Zhou Zhao, Wei Zhao, Xiaojun Chen, Jia Zhu, Lianqiang Zhou, and Zigang Cao. 2017. Personalized response generation via domain adaptation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1021–1024.

Jiaao Zhan, Yang Gao, Yu Bai, and Qianhui Liu. 2022. Stage-wise stylistic headline generation: Style generation and summarized content insertion. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4489–4495.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.

Yi Zhang, Tao Ge, and Xu Sun. 2020b. Parallel data augmentation for formality style transfer. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 3221–3228.

Yinhe Zheng, Zikai Chen, Rongsheng Zhang, Shilei Huang, Xiaoxi Mao, and Minlie Huang. 2021. Stylized dialogue response generation using stylized unpaired texts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14558–14567.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the International Conference on Computational Linguistics*, pages 1353–1361.