IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 February 2025, accepted 27 February 2025, date of publication 6 March 2025, date of current version 17 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3548967

RESEARCH ARTICLE

LLM-Based Text Style Transfer: Have We Taken a Step Forward?

MARTINA TOSHEVSKA, (Graduate Student Member, IEEE), AND SONJA GIEVSKA
Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, Skopje 1000, North Macedonia
Corresponding author: Martina Toshevska (martina.toshevska@finki.ukim.mk)

ABSTRACT Text style transfer is the task of altering the stylistic way in which a given sentence is written while maintaining its original meaning. The task requires models to identify and modify various stylistic properties, such as politeness, formality, and sentiment. With the advent of Large Language Models (LLMs) and their remarkable performances for a variety of tasks, numerous LLMs have emerged in the past few years. This paper provides an overview of recent advancements in text style transfer using LLMs. The discussion is focused on LLM-based approaches commonly used for text generation and their adoption for text style transfer. The paper is organized around three main groups of methods: prompting techniques for LLMs, fine-tuning techniques for LLMs, and memory-augmented LLMs. The discussion emphasizes the similarities and differences among the discussed methods and groups, along with the challenges and opportunities that are expected to direct and foster further research in the field.

INDEX TERMS Text style transfer, large language models, natural language processing, natural language generation.

I. INTRODUCTION

Text style transfer (TST) focuses on altering specific stylistic attributes of text while preserving its underlying content. This task involves transforming texts to meet various stylistic criteria, such as sentiment, formality, or politeness while preserving their explicit meaning. Adjusting the style could be crucial in improving communication and/or writing skills by making the text more polite or more suitable for formal occasions. In the realm of social media platforms or online communities, transferring the style could enhance user interactions and reduce misunderstandings by adjusting the emotional tone of user content to make it more positive or to remove inappropriate language.

Prior to the advent of Large Language Models (LLMs), methods for TST relied mostly on encoder-decoder architecture [1], generative adversarial networks [2], and reinforcement learning [3]. Considering the nature of the task, an ideal approach would be the use of large parallel corpora in a supervised manner. Since such data is difficult to construct and not always available, most of the methods were focused on unsupervised learning with non-parallel datasets often relying on approaches that include additional input [4], [5] or components [6], [7] and use composite learning objectives [8] or rewards [9] to facilitate text style transfer.

The lack of large and high-quality parallel datasets for diverse styles, which limits the performance of existing approaches, remains a big challenge. The existing approaches face difficulties in disentangling style from content since separating the writing style from the main content is still difficult. Balancing the trade-off between preserving the original meaning and successfully modifying the style remains a significant challenge, as adjusting the style sometimes leads to unintended changes in the meaning and/or a reduction in fluency. The advanced capabilities of LLMs may contribute to improving the text style transfer task by using their pre-trained knowledge and ability to incorporate external knowledge.

With the advent of LLMs, their application for TST gained attention primarily focused on exploring prompting techniques [10], [11]. By following task-specific instructions and/or examples, prompting enables a flexible and efficient method to explore various styles, for a wide range of text style transfer tasks, including arbitrary styles that often

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

lack extensive parallel data. While prompting techniques significantly reduce the need for huge datasets, other methods such as fine-tuning [12], and memory augmentation [13] have also motivated recent approaches. Fine-tuning enables adapting to specific stylistic properties using smaller and more targeted datasets, especially for specialized styles where general models often underperform. Memory augmentation allows the incorporation of external information and/or internal memories to improve the generation. It can help the model generate text that better aligns with the desired target style and maintains the original content.

This paper provides an overview of recent advancements in text style transfer using LLMs as exploratory research into the strengths, weaknesses, and future possibilities. In an attempt to build a current view of the research in the field of TST, we have distilled a list of papers from the current body of literature on TST, focusing on state-of-the-art (SOTA) LLM-based approaches. Beginning with a brief introduction to the tasks, datasets, evaluation methods, and SOTA pre-LLM approaches we continue the discussion of the LLM-based approaches for text style transfer. We further discuss these methods grouped among three main categories: *prompting techniques*, *fine-tuning techniques*, and *memory-augmented LLMs*. Pre-LLM approaches were selected based on their usage as a benchmark for comparison with other methods.

By the time of writing this paper, there were a few surveys covering different aspects of text style transfer [14], [15], [16], [17], [18], and many surveys summarizing different aspects and use cases of LLMs [19], [20], [21], [22], [23], [24]. However, none of them were focused particularly on the application of LLMs for text style transfer.

The rest of the paper is organized as follows. After the introductory section, Section II provides an overview of the text style transfer tasks, datasets, and pre-LLM approaches. Section III gives a brief introduction to techniques for LLM-based text generation that is followed by a discussion of LLM-based approaches for text style transfer in Section IV and its challenges in Section V and implications in Section VI. Section VII concludes the paper.

II. TEXT STYLE TRANSFER

A. OVERVIEW OF TEXT STYLE TRANSFER: TASKS, DATASETS, AND EVALUATION

Text style transfer is the task of altering and/or adapting the stylistic way in which a sentence is written. It involves generating a new sentence that preserves the same explicit meaning but stylistically differs from the original one. The term style could refer to diverse properties such as the individual style of the author, politeness, formality, sentiment, and other styles. Depending on the style property that is altered several text style transfer tasks could be identified. Modifying the level of politeness or formality is known as *politeness transfer* and *formality transfer*, respectively. Adapting the emotions manifested in a sentence is known as *sentiment transfer*. Rewriting text to match an individual author's style falls under *personal style transfer*. An example is Shakespearean writing style.[1] *Expertise style transfer* is the task of transforming a sentence to make it more understandable for non-experts in a particular field (e.g. medical experts vs. layman). Correcting the implicit and potentially undesirable bias is referred to as *neutralizing subjective bias*. Substituting the offensiveness with a neutral style is the objective of the *transferring offensive to non-offensive text* task. Transforming a sentence that contains toxic speech into a neutral one is the task of *detoxification*.

TST comes down to a sequence-to-sequence problem for which having a parallel dataset would be preferable. A parallel dataset is composed of sentence pairs with the same meaning written in two different styles. Depending on the style attribute, various parallel datasets have been used in the previous research: **GYAFC** [25] (*formality transfer*), **Shakespeare** [26], [27] (*personal style transfer*), **WNC** [28] (*bias correction*), **ParaDetox** [29] (*detoxification*), and others. Obtaining and labeling large parallel datasets that are essential, especially for larger models is often unfeasible, resulting in increased attention on methods that do not require the use of a parallel dataset. Non-parallel datasets contain sentences in different styles, however, they are not mapped between styles so there is no sentence with the same meaning in another style. The most frequently used non-parallel datasets include: **Yelp**[2] (*sentiment transfer*), **IMDB** [30] (*sentiment transfer*), **Amazon** [31] (*sentiment transfer*), and others. The creation of pseudo-parallel data for supervised training on non-parallel datasets has also been studied [32].

Evaluating TST is a non-trivial challenging task because of the intertwined nature of semantic and stylistic properties in language. It is often difficult to alter one without affecting the other. Given the difficulties of conducting human evaluation, automatic evaluation was more often performed. The evaluation is mostly focused on three aspects: (1) quality of semantic content preservation, (2) quality of style, and (3) fluency. Evaluation metrics that measure the extent to which the generated sentence matches human output are used to evaluate the quality of semantic content preservation. The most common evaluation metrics are **BLEU** [33], **METEOR** [34], **ROUGE-L** [35], **BERTScore** [36], and others. BLEU is often used to measure the degree to which the model directly copies the input sentence. This variation is called self-BLEU (**sBLEU**), while the distance from the ground-truth references is referred to as reference-BLEU (**rBLEU**). Quality of style is calculated as a percentage of generated sentences labeled with the target style by a pre-trained style classifier i.e. **accuracy**, while fluency is often assessed by computing the **perplexity** with a pre-trained language model. The most frequently used pre-trained language model for this purpose is GPT-2 [37]. Several metrics such as **joint score** [38], **geometric** and **harmonic**

[1] https://en.wikipedia.org/wiki/Shakespeare's_writing_style, last visited: 05.08.2024

[2] https://www.yelp.com/dataset, last visited: 05.08.2024

mean [39] were applied to combine the three aspects into a single metric.

B. PRE-LLM APPROACHES FOR TEXT STYLE TRANSFER

Before the rise of LLMs, TST methods were primarily based on the encoder-decoder architecture [1], Generative Adversarial Networks (GANs) [2], and Reinforcement Learning (RL) [3]. The methods based on encoder-decoder [4], [5], [6], [40], [41], [42], [43] consisted of an encoder for creating a latent style-free representation and a decoder for generating a sentence in the target style. Some of the methods [6], [42], [43] included an additional component (most frequently a style classifier) to assist the generation further. **DeleteOnly** [4] and **DeleteAndRetrieve** [4] used GRU [44] for both the encoder and decoder. Apart from the input sentence, as an additional input, **DeleteOnly** used an encoded representation of the target style, while **DeleteAndRetrieve** used an embedded representation of the style markers[3] of the most similar sentence from the corpus of sentences with the target style. **B-GST** [5] and **G-GST** [5] were built with the same approach, but they used Transformer-based decoders instead of GRU. StyleTransformer [42] utilized an additional Transformer encoder to distinguish the style of the generated sentences. Similarly, **StableStyleTransformer** [6] applied a CNN [45] classifier to classify the style of the generated sentence.

GAN-based approaches [7], [8], [46], [47] for TST exploit adversarial training. A generator was utilized for generating a sentence in the target style. These approaches replace the style classifier component from the previous group with a discriminator component trained with adversarial loss that aims to distinguish between real and generated sentences in the desired styles. **AAE** [46] aligned posterior probability distributions obtained from the encoders for both input and target styles with a feed-forward discriminator. **CAAE** [46] used two CNN discriminators with the same objective. **MultiDecoder** [7] and **StyleEmbedding** [7] trained a discriminator to determine the style according to the encoded representation of the input sentence with two adversarial goals: maximizing the probability and the entropy of correctly predicting the style. To better capture the desired style, **StyleDiscrepancy** [8] included an additional discriminator and loss function that was responsible for assessing if the sentence matches the target style.

RL-based approaches [9], [48], [49], [50] employ a reward-based system to generate a sentence in the desired style. **DualRL** [9] designed a two-fold reward: a reward for changing the style and a reward for preserving the content. The reward for changing the style measures how well the generated sentence matches the target style and is computed with a style classifier, while the reward for preserving the content measures the percentage of the content that was preserved with the reconstruction of the input sentence. An LSTM-based [51] encoder-decoder was trained with the policy gradient algorithm [3] to maximize the expected two-fold reward. **DRL** [50] used self-attention scores of a trained style classifier as a reward for changing the style and BLEU [33] computed only with content words as a reward for preserving the content. A fluency reward was introduced making the overall reward three-fold. This reward was calculated as a token-level likelihood obtained with a GPT-based [52] language model.

[3]Style markers are words that have the most discriminative power for determining the style of a sentence.

III. LLM-BASED APPROACHES FOR TEXT GENERATION

Large Language Models (LLMs) demonstrated remarkable abilities in various NLP tasks by leveraging the Transformer architecture [53]. Recently proposed methods employ techniques such as zero-shot, few-shot, and Chain-of-Thought (CoT) prompting [54], [55], [56] to obtain task-specific outputs. Fine-tuning techniques and their parameter efficient variants, such as Parameter Efficient Fine-tuning (PEFT) [57] and Prefix Tuning [58] focus on adapting the models for particular tasks. By incorporating external knowledge in the generation process, Retrieval Augmented Generation (RAG) [59] techniques aim to enhance the performance. Knowledge Augmentation (KA) techniques enhance the performance of knowledge-intensive tasks [60], [61]. We encourage the reader to refer to the Appendix for a more detailed description of these techniques.

IV. LLM-BASED APPROACHES FOR TEXT STYLE TRANSFER

In this section, we describe several approaches for LLM-based text style transfer laid out through the lenses of three main categories: *prompting techniques for LLMs*, *fine-tuning techniques for LLMs*, and *memory-augmented LLMs*. Prompting methods are one of the earliest explored methods that include standard zero-shot and few-shot prompting, selecting optimal prompt via routing, text editing via prompting, as well as using LLMs as auxiliary components to enhance traditional pre-LLM approaches. Fine-tuning involves adapting to specific tasks through standard fine-tuning, instruction fine-tuning, parameter-efficient fine-tuning, and reinforcement learning. Memory-augmented LLM approaches, though relatively unexplored, focus on integrating external knowledge as part of the input prompts and aligning models with dynamic attribute graphs. The categorization of the LLM-based text style transfer methods in groups and subgroups is displayed in Figure 1. The specific text style transfer approaches within each category, along with the LLMs used and the TST tasks they address, are presented in Table 1 and discussed in the following subsections.

A. PROMPTING TECHNIQUES FOR TEXT STYLE TRANSFER

Prompting techniques are one of the first approaches that were explored for LLM-based text style transfer. The approaches include standard zero-shot and few-shot prompting techniques that rely on input examples to guide

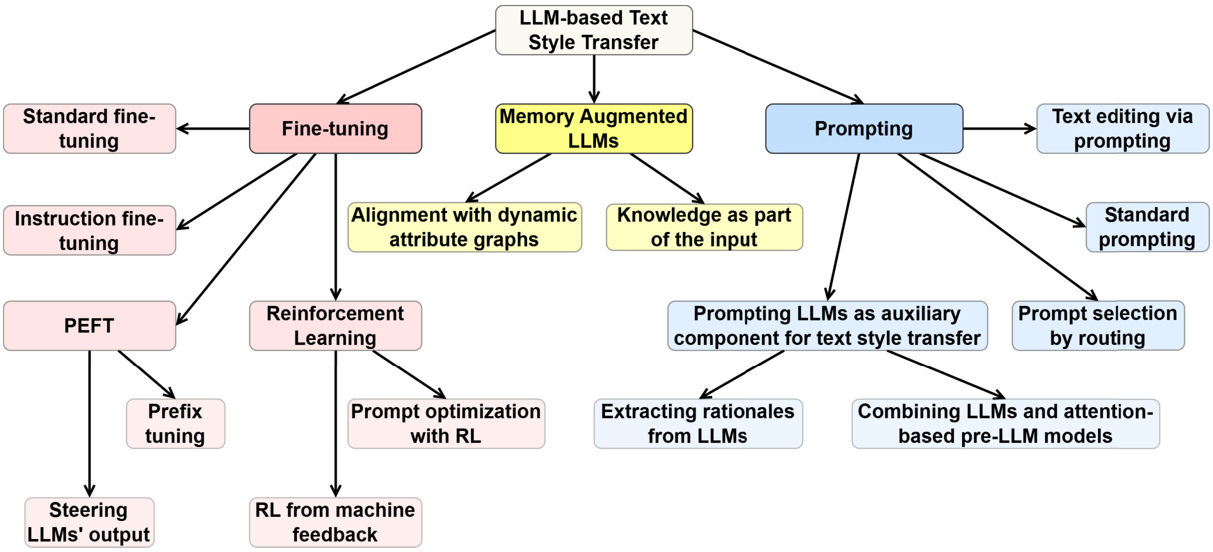

FIGURE 1. Taxonomy for the techniques and methods for LLM-based text style transfer discussed in this paper.

the generation of a sentence in the desired target style. Prompting strategies that utilize text editing are focused on defining prompts to guide the model to edit the input sentence. Methods based on prompt routing use a routing mechanism to select the best prompt that is the most effective for the particular task. Additionally, LLMs can be used as an auxiliary component to enhance other pre-LLM-based models. A general architecture of the prompting methods is shown in Figure 2 a).

1) STANDARD ZERO-SHOT AND FEW-SHOT PROMPTING

Standard zero-shot and few-shot prompting techniques leverage a few or no task-specific demonstrations to guide the LLM in generating text in the desired target style. Several methods have been proposed [10], [11], [69], [77]. Few approaches explore the performances of zero-shot and few-shot techniques [10], [11], [69]. **Augmented zero-shot** [10] used a single set of exemplars as part of the prompt, that included a variety of sentence rewriting operations instead of exemplars specific to the target style. Apart from the vanilla prompt, which only specifies the target style, **Prompt-and-Rerank** [11] explored a contrastive prompt (specifies both the source and the target style to create a clear contrast between them) and two negation prompts (specify the target style as a negation of the source style and vice versa). To obtain the output sentences, this model generated K candidate outputs, then ranked them according to 3-fold criteria (fluency, transfer strength, and textual similarity) and took the highest ranking one as the final output.

The experimental results showed that smaller models struggle, especially for zero-shot prompting [69]. GPT-2-small [11] often copied long sections from the input. Larger models performed better [10], [11]. However, the improvement diminished with increasing the size [69]. Increasing the model size from 1B to 3B showed a significant performance boost, but further increasing the size did not provide such an increase in performance. Augmented zero-shot [10] demonstrated low BLEU scores because of the tendency to add additional information to the generated sentences. Overall, few-shot prompting improved performance over zero-shot prompting [69]. Specific syntax and semantics of the prompt template were shown to significantly impact model performance [11]. The use of contrastive prompts and re-ranking was shown to improve style quality.

2) TEXT EDITING VIA PROMPTING

Text editing via prompting involves defining a prompt that guides the model to edit a specific part of the input sentence in order to generate an output sentence that conforms to the desired target style. Two of the recently proposed methods rely on text editing via prompting for text style transfer. **PEGF** [77] guided the LLM to modify only a small part of the sentence within the editing region. It relied on prompting for two steps. An LLM was first prompted to identify stylistic words by assigning each word a score in the range $[-1, +1]$. Words with a score higher than a predefined threshold were considered stylistic words. The LLM was again prompted to "edit" the stylistic words via implicit (stylistic words were specified in the prompt) or explicit (each stylistic word was replaced with a "[MASK]" token) masking. To be more controllable and not suffer from error accumulation, **PromptEdit** [39] transformed the text style transfer task into a classification problem. Given a candidate sentence, the goal was to obtain a style score with an LLM. An edit-based search was applied using the steepest-ascent hill climbing (SAHC) algorithm [98] for local search using editing operations: insertion, deletion, and replacement. For each editing position, every editing operation was performed

TABLE 1. LLM-based text style transfer approaches.

Approach	Year	LLMs	TST Task(s)	LLM-based TST Approach
Augmented zero-shot [10]	2022	GPT-3 [62], LaMDA [63]	Formality Transfer Sentiment Transfer	Standard prompting
Prompt-and-Rerank [11]	2022	GPT-2 [37], GPT-J [64], GPT-Neo [65]	Formality Transfer Sentiment Transfer	Standard prompting
LowResBART [12]	2023	BART [66]	Sentiment Transfer	Standard fine-tuning
ProSwitch [13]	2024	LLaMA2-Chat [67]	Expertise style transfer	External knowledge as part of the input
APR [38]	2024	GPT-2 [37], GPT-J [64], GPT-3.5 [68]	Formality Transfer Sentiment Transfer Personal Style Transfer	Prompt selection by routing
PromptEdit [39]	2023	GPT-J [64]	Formality Transfer Sentiment Transfer Personal Style Transfer	Text editing via prompting
Prompt&Fine-tune [69]	2024	GPT-3.5 [68], Falcon [70], LLaMA [71], Mistral [72], BLOOM [73], ChatGLM [74], OPT [75], Zephyr [76]	Sentiment Transfer Text Detoxification	Standard prompting, Standard fine-tuning
PEGF [77]	2024	GPT-3.5 [68]	Formality Transfer Sentiment Transfer	Text editing via prompting, RL from machine feedback
CoTeX-TA, CoTeX-TB [78]	2024	PaLM2 Unicorn [79]	Formality Transfer Personal Style Transfer Text Detoxification	Extracting rationales from LLMs
LLM-as-signal, Prompt-then-AM, AM-then-prompt, AM-as-demo [80]	2024	BART [66], ChatGLM2 [74]	Sentiment Transfer	Combining LLMs and attention-based pre-LLM models
SubReddit-T5, SubReddit-BART [82]	2023	BART [66], T5 [81]	Style Transfer to community-specific SubReddit language	Standard fine-tuning
CycleGAN-T5, CycleGAN-BART [83]	2024	BART [66], T5 [81]	Formality Transfer Sentiment Transfer	Standard fine-tuning
TEP, CEP [84]	2024	GPT-3 [62], GPT-3.5 [68], Falcon [70], LLaMA [71], Mistral [72], Zephyr [76], GPT-4 [85], and other LLMs	Personal Style Transfer	Instruction fine-tuning
PrefixTune [86]	2023	GPT-2 [37]	Sentiment Transfer	PEFT
StyleVectors [87]	2024	Alpaca [88]	Sentiment Transfer Personal Style Transfer	PEFT
sNeuron-TST [89]	2024	LLaMA-3 [90]	Formality Transfer Sentiment Transfer Personal Style Transfer Text detoxification	PEFT
RLPrompt [91]	2022	GPT-2 [37]	Sentiment Transfer Personal Style Transfer	Prompt optimization with RL
PPO [92]	2024	GPT-J [64], LLaMA [71], BLOOM [73], OPT [75]	Rewriting inappropriate argumentation	RL from machine feedback
T5-KA [93]	2024	T5 [81]	Formality Transfer	External knowledge as part of the input
SKGPrompt [94]	2025	GPT-J [64], GPT-Neo [65], LLaMA-2 [67], T5 [81], FLAN-T5 [95]	Formality Transfer Personal Style Transfer Text Detoxification Neutralizing subjective bias	External knowledge, as part of the input
DATG [96]	2024	LLaMA-2 [67], Falcon [70], BLOOM [73], Alpaca [88], Phi-2 [97]	Sentiment Transfer Text Detoxification	Alignment with dynamic attribute graphs

[]

and the candidate sentence with the highest score was selected. The average edit distance was 2.9 and 4.7 steps for sentiment transfer and formality transfer, respectively.

Methods based on editing achieved high self-BLEU scores because of the preservation of the input content [77] but yielded a better balance between content preservation and style transfer strength with 20x smaller LLMs [39]. Content preservation improved when more samples were included in the prompt, but style transfer strength and fluency decreased [77]. Prompt editing [39] achieved better performances than Prompt-and-Rerank [11], and a better balance between content preservation and style transfer strength compared to the Augmented zero-shot [10]. The improvement was smaller for the formality transfer task because it is more challenging than the sentiment transfer task.

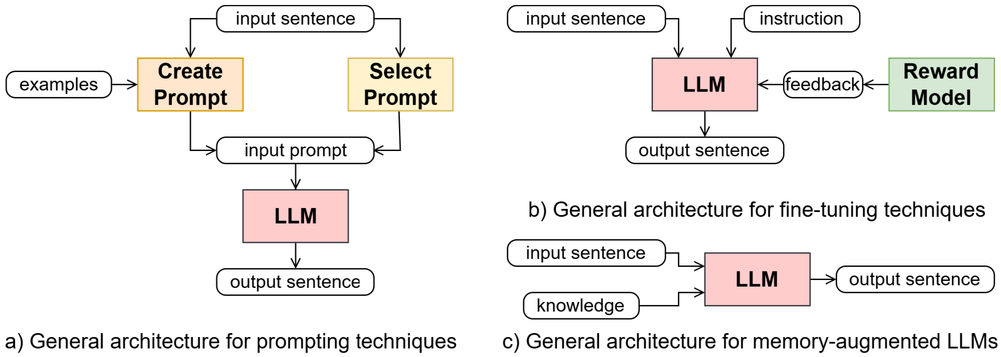

FIGURE 2. Architecture for different LLM-based approaches for text style transfer. a) General architecture for prompting techniques. b) General architecture for fine-tuning techniques. c) General architecture for memory-augmented LLMs.

3) PROMPT SELECTION BY ROUTING

Routing mechanisms identify the most effective prompts for a given task. By evaluating multiple options, these mechanisms ensure that the LLM receives the most effective instructions for the task. **APR** [38] utilized a prompt router to determine which prompt among a few predefined seed prompts with different degrees of detail about the task is optimal for the input sentence. The prompt router was trained in a supervised manner by scoring a subset of the sentences paired with each of the seed prompts. According to the style transfer score (considering content preservation and quality of style), the pairs of seed prompts and sentences were separated into positive and negative samples. To generate the output sentence in the target style, first, the optimal prompt was selected with the prompt router and then the LLM was prompted with it. This approach generated outputs with better style transfer strength and more fluent and grammatical text. Compared with the random selection of a prompt, the performance on prompt selection based on style transfer score was significantly better.

4) PROMPTING LLMS AS AUXILIARY COMPONENT FOR TEXT STYLE TRANSFER

LLMs can act as auxiliary components to complement and enhance pre-LLM-based models. Despite most methods being focused on utilizing LLMs directly for text style transfer, several methods explored the option of using the LLM prompting techniques in a way that enhances the traditional text style transfer approaches [78]. **CoTeX-TA** [78] and **CoTeX-TB** [78] followed the CoT technique for improving text style transfer by extracting rationales from an LLM. CoT was applied in two distinct methods: Target-Blind (TB) in which only the source text and the target style were specified, and Target-Aware (TA) with the corresponding human-annotated target text specified additionally. The generated data was used to fine-tune smaller student models. A similar approach, **LLM-as-signal** [80], was applied to fine-tune a mask predictor to identify and modify stylistic words with training data generated by LLM. **Prompt-then-AM** [80] used prompting to obtain a sentence with the target style and then applied attention masking to identify and modify stylistic words. **AM-then-prompt** [80] did the opposite i.e. first applied attention masking to edit the original text and then rewrote the prediction by prompting the LLM. **AM-as-demo** [80] explored few-shot prompting with samples generated with attention masking.

Fine-tuning with samples generated following CoT surpassed previous unsupervised, few-shot prompting and instruction fine-tuned methods for formality transfer and achieved comparable performance on the text detoxification task [78]. The approach also showed an advantage in low-resource settings possibly due to the high quality of data generated by the LLM. However, including target output sentences did not exceed the performance [78]. On the contrary, using LLM-generated outputs as demonstrations for editing stylistic words showed a negative impact because the generated sentences had poor style strength [80]. The opposite, i.e. utilizing samples generated with attention masking for editing stylistic words helped the LLM to learn the target styles better. LLMs were more likely to produce more fluent output sentences, while attention masking methods performed better in transforming sentences to the target style [80], therefore applying prompting followed by attention masking effectively balanced the fluency and style strength.

B. FINE-TUNING LLMS FOR TEXT STYLE TRANSFER

Fine-tuning techniques for LLM-based text style transfer were used to adapt to specific TST tasks. Standard fine-tuning adjusts LLM weights using task-specific datasets, while instruction fine-tuning further enhances the LLM with task-specific instructions. Parameter-efficient methods, such as prefix tuning, learn a sequence of task-specific vectors without altering the base model weights. Additionally, representation fine-tuning methods adjust the internal state of LLMs to steer the generation toward the desired target style.

RL-based methods, such as RL from machine feedback to verify the validity of the generated output and policy networks to find an optimal prompt, were also explored for text style transfer. A general architecture of the fine-tuning methods is shown in Figure 2 b).

1) STANDARD FINE-TUNING AND INSTRUCTION FINE-TUNING

Several standard fine-tuning approaches were explored for text style transfer with task-specific datasets [12], [69], [82], [83]. **LowResBART** [12], **SubReddit-T5** [82], **SubReddit-BART** [82], **CycleGAN-T5** [83], and **CycleGAN-BART** [83] relied on the standard fine-tuning approach on a pre-trained LLM. LowResBART was fine-tuned in multiple experiments, each including one or a combination of the following methods: hyperparameter tuning, adding explicit guidance via prompts, self-training using synthetic data generated by the model, use of paraphrasing as a data augmentation technique, and incorporating a reward (obtained from a style classifier) into the training loss. SubReddit-T5 and SubReddit-BART were fine-tuned to complete a prompt with provided examples, to follow the style of those examples. CycleGAN-T5 and CycleGAN-BART utilized a fine-tuning approach based on CycleGAN [99] for text style transfer to apply self-supervision at the sequence level to handle mixed-style inputs. On the other hand, **TEP** [84] and **CEP** [84], evaluated the instruction-based fine-tuning using a four-part prompt that specified the task, instructions, output indicator, and example text in the target style (the experiments were only focused on particular author styles). TEP provided task definition with a short snippet of author text, while CEP added additional author data or guided instructions in the form of a few key linguistic features that potentially demonstrate the author's unique linguistic preferences.

Fine-tuning proved to achieve the highest performance gains with a strong performance for most of the LLMs including the ones whose performances on zero-shot and few-shot prompting were weak [69], [83], but further fine-tuning on more epochs showed to worsen the performances [82]. Hyperparameter tuning also improved the performance [12]. Adding prompt guidance and incorporating style rewards into a training loss improved style accuracy [12] while self-training on synthetic data improved content preservation. However, increasing the number of synthetic samples did not provide much improvement due to an imbalance between real and synthetic samples [12], but adding more data about the target style[4] in the prompt led to performance improvement [84]. For a subset of TST tasks, instruction-tuned and chat-based models showed better performance than their base variants in zero-shot and few-shot settings [69]. An example of the opposite is the task of generating text in a language specific to a particular SubReddit community. Fine-tuning yielded worse results than the zero-shot setting and the model tended to simply copy the source text [82].

[4]More data about the target author style.

2) PARAMETER EFFICIENT FINE-TUNING (PEFT)

For text style transfer, PEFT was explored in two directions i.e. prefix tuning and adjusting the LLM internal state to steer the generation toward the desired target style. **Prefix-Tune** [86] relied on the prefix tuning approach incorporating three distinct prefixes: style prefix to encode the target style, content prefix to extract content information, and shared prefix to provide task-related information. The model followed adversarial training with both the generator and the discriminator being prefix tuning-based LLMs. The generator was trained with a combination of self-reconstruction loss, style transfer loss, and cycle generation loss. The discriminator was trained with a negative log-likelihood of predicting the style of the generated sentence or determining whether it was fake. **StyleVectors** [87] and **sNeuron-TST** [89] focused on modifying the internal state of the LLM to guide the generation i.e. steering its output. StyleVectors experimented with two types of style vectors that were added to the activations of the hidden layers during the generation. Training-based style vectors were created by learning an individual steering vector for each target sentence. Activation-based style vectors were created by aggregating activations of chosen layers recorded during the forward pass of the input. sNeuron-TST identified source-style and target-style neurons associated with source and target styles, respectively. Neurons with zero activations were selected as style neurons and associated with the corresponding style. The output sentence was generated with the LLM where the source-style-only neurons were deactivated to enhance the stylistic diversity and give a higher probability of target-style words.

Prefix tuning improved content preservation and fluency [86] and provided results that were close to fine-tuning the whole model. Regarding the different prefixes, there was a drop in content preservation without the content prefix and a drop in both style control and content preservation without the shared prefix. Without the style prefix, the LLM directly copied the input sentence, failing to modify the style. Steering the output of the LLM with style vectors enabled a continuous and adjustable modulation of the LLMs' outputs and provided a smoother transition than prompting-based approaches. It achieved superior style transfer accuracy and fluency in comparison with baselines [87]. However, deactivating source-style neurons improved the accuracy, but decreased the fluency because of the deactivation's impact on the word distribution [89]. Deactivating target-style neurons decreased both accuracy and fluency. Comparing the two style vector types, training-based style vectors were more expensive in terms of performance and resources than activation-based style vectors [87]. According to the analysis of neuron activations and the determination of the style-specific words, the latter layers of the model were shown to be responsible for the style-related outputs [89].

3) APPROACHES BASED ON REINFORCEMENT LEARNING

Verifying the validity of the generated output and finding optimal prompts via policy networks were the two directions explored for LLM-based text style transfer with RL. **PEGF** [77] applied RL from machine feedback to verify the validity of the detected editing region of the input sentence. It used a classifier as a discriminator to compute a score for the input sentence and the masked input sentence in which the stylistic words were replaced with a special "[MASK]" token. Then the results were fed back to the LLM. **RLPrompt** [91] utilized RL for discrete prompt optimization by training a policy network to generate the desired prompts. The agent selected prompt tokens one by one according to a policy to maximize the reward, at each time step given the previous tokens. For text style transfer the reward was a sum of the content preservation score (computed via matching token embeddings similar to BERTScore) and style transfer strength score (computed with a BERT-based style classifier). A similar reward model was used by another RL from machine feedback approach [92] that applied PPO [100] to obtain a better version of an initial policy. An initial policy was obtained via various prompting techniques that solve the task to a certain extent. Several policies were learned, each with a different weight for a balance between content preservation and style transfer strength.

Using a discriminator and its feedback for validating editing regions showed a significant impact on the performance since, without it, assessing the correctness of the edited regions generated by the models was not possible [77]. Employing RL to select prompt tokens was shown to improve performance significantly over prompting baselines. These prompts performed better on average with lower variance than manual prompts [91]. By comparing different model sizes, the performance monotonically increased from smaller to larger models [91]. Larger models showed slightly lower content preservation and style accuracy, but higher perplexity indicating that the approach preserved the LLM fluent capabilities [91]. Instruction fine-tuned LLaMA demonstrated the best overall performance and applying PPO to the LLaMA managed to align to those performances, with the best performance achieved by an equal balance between the two reward values [101].

C. MEMORY-AUGMENTED LLMS FOR TEXT STYLE TRANSFER

Memory augmentation for LLM-based text style transfer is a relatively unexplored area with current approaches primarily focused on integrating external knowledge as part of the input prompt and alignment with dynamic attribute graphs. A general architecture of the memory-augmented LLM-based methods is shown in Figure 2 c). Fine-tuning with knowledge augmentation by including additional knowledge as part of the input sentence and/or instruction was explored by **ProSwitch** [13], **T5-KA** [93] and **SKGPrompt** [94]. ProSwitch explored PEFT and full fine-tuning with knowledge-enriched instruction that included article snippets to provide implicit knowledge of the target style. T5-KA included the sentence from the set of sentences in the target style that is most similar to the input sentence, as part of the input to the LLM. SKGPrompt explored prompting augmented with information extracted from a style knowledge graph that contained words and semantic relations between them. Semantic relations for the top three style markers were extracted from the style knowledge graph and added to the input sentence. **DATG** [96] employed dynamic attribute graphs to identify keywords aligned or opposed to the target style. The text sequences were transformed into a directed weighted graph informed by classifier scores for alignment with the target style. Two graphs, positive and negative, were created. The output sentences were generated with two strategies. The Logits-Boost strategy influenced token probabilities associated with positive and negative nodes by adjusting nodes in the generation algorithm (**DATG-L**). The Prefix-Prompt strategy guided the LLM towards highlighting positive nodes and avoiding negative nodes by appending particular prompt prefixes (**DATG-P**).

Knowledge-enriched instruction provided more detailed guidance and improved the ability to switch styles [13] while augmenting with the most similar sentence achieved better performances than standard fine-tuning and zero-shot prompting [93]. Augmenting with information extracted from a style knowledge graph improved content preservation possibly due to the structure of the prompt that provided specific word choices to guide the generation [94]. In comparison with PEFT techniques, standard fully fine-tuned models generated longer sentences with more reasoning steps and fewer technical terms [13] suggesting that they learned expression better than wording. Analyzing the effect of model size showed that increasing the model size improved the performance as suggested with the T5-base model achieving better evaluation results than the T5-small model [93]. Logits-Boost strategy showed superior performance, while Prefix-Prompt had weaker performance [96] with higher relevance scores likely due to the keywords in the prompt that acted like anchors. Both approaches effectively reduced toxicity and controlled the sentiment, despite demonstrating varying performance in both directions for sentiment transfer.

V. CONCLUDING REMARKS ON LLM-BASED TST: CAPABILITIES AND LIMITATIONS

Before concluding, it is important to highlight the potential of LLMs for text style transfer. Although significant progress has been made in recent years, many challenges and opportunities remain to be explored.

A. PROMPT AND INSTRUCTION DESIGN FOR LLM-BASED TEXT STYLE TRANSFER

Designing prompts and instructions plays a significant role in text style transfer. Human intuition is to design fluent prompts that clearly describe the task with explicit directions.

While prompts are often designed in a way to make sense for a human, one research has shown that prompts that seem nonsensical or differ from what a human would consider optimal i.e. are "gibberish" can lead to better performance [91]. For example, one optimal prompt learned for sentiment transfer was *"Parameters Comparison) = (Compare either"*. Learning optimal prompts may be challenging, especially for larger models that require more resources. Prompts learned on smaller models demonstrated similar or better performance when applied on larger models [91] thus highlighting the potential of leveraging smaller models for learning optimal prompts more efficiently and then using them for larger models to benefit from their capabilities.

B. LLM-BASED TEXT STYLE TRANSFER

The performance of LLM-based text style transfer methods varies depending on the size of the model, the complexity of the task, and the approach used. Larger models demonstrated better performance in generating sentences in the desired target style [10], [11], [91], [93], [94], while smaller models faced notable challenges, particularly for zero-shot prompting [69]. However, further increasing the model size was shown to diminish the improvement in performance [69] suggesting that increasing the model size is not the general solution. Selecting the optimal prompt with RL improves the performance in comparison with prompting baselines [91]. Additionally, methods based on prompt routing demonstrated better style transfer strength and fluency compared to standard prompting, as they select the most suitable prompt for the given task [38]. Contrastive prompts have also shown improvements in style quality by creating a clearer distinction between the source and target style in the generated text [11].

Providing task-specific examples with few-shot prompting offers an improvement over zero-shot prompting approaches. When compared to prompting techniques, fine-tuning achieved superior performance [69]. On the other side, fine-tuning across additional epochs led to worse performances for some text style transfer tasks such as generating sentences in the style used in a particular SubReddit community [82]. As a parameter-efficient alternative, prefix tuning provided results close to the full fine-tuning of the model [86], while also improving content preservation and fluency. Additionally, incorporating style rewards into training objectives was shown to enhance style transfer strength by explicitly guiding the model toward stylistic modifications [91]. Self-training on synthetic data generated by LLMs further improved content preservation [12].

The underlying text style transfer task and its complexity shape the overall performance of the model. For example, the improvement achieved with text editing via prompting, in comparison with standard prompting approaches, was smaller for formality transfer than sentiment transfer [39] since formality transfer is a more challenging task. Methods based on prompt editing achieved high self-BLEU scores because they only modify small parts of the sentence while preserving most of the input structure, leading to strong content retention [39]. Moreover, augmenting input prompts with information extracted from a style knowledge graph has further improved content preservation by providing structured contextual guidance [94]. Additionally, fine-tuning with LLM-generated samples outperformed pre-LLM unsupervised, few-shot prompting, and instruction fine-tuning methods for formality transfer, but achieved comparable results for text detoxification [78].

C. PROMPTING VS. FINE-TUNING VS. MEMORY-AUGMENTED LLMS FOR TEXT STYLE TRANSFER

Fine-tuning generally improves the performance of text style transfer models compared to prompting, as it allows the model to adapt more closely to the desired target style by updating its parameters with supervised examples. This additional adaptation results in higher quality outputs, particularly for complex tasks such as generating text in a particular person's style. However, the effectiveness of fine-tuning depends on the quality and size of the training data, as well as the specific style transfer task. In cases where the available training data does not fully capture the nuances of a particular style, prompting may still be a viable alternative.

Despite its performance advantages, fine-tuning is computationally expensive and requires significant training time, particularly for larger models. The process involves updating model weights through multiple training epochs, demanding substantial hardware resources, which can be prohibitive for researchers or practitioners with limited computational capacity. In contrast, prompting methods do not require modifying the model parameters and are more cost-effective. By leveraging pre-trained knowledge, prompting allows for flexible adaptation without the need for extensive retraining, making it a practical choice for many real-world applications.

Memory augmentation offers another alternative that surpasses both prompting and fine-tuning in performance by allowing the model to retrieve and utilize external knowledge during inference. This method provides access to additional context and factual information, enabling the model to produce outputs that are more accurate and contextually appropriate. Unlike fine-tuning, which integrates knowledge directly into the model weights, memory augmentation keeps external knowledge sources separate, reducing the risk of catastrophic forgetting and improving generalization across different tasks. However, the effectiveness of memory augmentation depends on the quality and relevance of the external data used for retrieval.

The choice between these approaches is further influenced by the trade-off between model size and computational efficiency. Larger models generally achieve superior performance in text style transfer tasks due to their increased capacity to capture linguistic patterns and stylistic nuances. However, this improvement comes at the cost of higher computational demands, longer inference times, and greater energy consumption. In contrast, smaller models require significantly fewer resources but tend to struggle with complex transformations, particularly in zero-shot and few-shot scenarios. As a result, balancing model size, adaptation strategy, and computational feasibility remains a crucial consideration for optimizing text style transfer performance.

D. LLM-BASED APPROACHES VS. PRE-LLM APPROACHES FOR TEXT STYLE TRANSFER

In comparison to traditional pre-LLM state-of-the-art models, although achieving comparable performances [69], LLMs offer less fine-grained control in the properties of the style-transferred text than methods that see task-specific training data [10]. This was manifested by not improved [12] or lower [91] content preservation, and variable style strength. Several approaches demonstrated a notable improvement in fluency by generating smoother outputs [77] thus suggesting that LLM-based approaches preserve their fluent generation capabilities [91]. On the contrary, pre-LLM approaches had relatively high style strength, but low fluency [80]. Nonetheless, the performance of LLMs versus non-LLM-based methods depends heavily on the specific text style transfer task as different methods showed varying positive or negative improvements on different tasks [78].

Traditional pre-LLM methods rely heavily on large (parallel) datasets that are difficult to create and not always available. By leveraging zero-shot or few-shot capabilities, LLMs perform text style transfer with no or few examples thus reducing the need for huge (parallel) datasets. This ability also allows transferring the sentences to arbitrary and unseen styles such as "more comic" or "more melodramatic" [10]. Output sentences generated by LLMs demonstrated strong fluency, thus making them beneficial for creating synthetic data for supervised training of smaller or non-LLM models [80].

E. LIMITATIONS AND OPEN CHALLENGES OF LLM-BASED APPROACHES FOR TEXT STYLE TRANSFER

Although LLMs are capable of generating diverse text for many tasks they still face limitations. Sometimes, the obtained output cannot be parsed into usable answers [10] or it may not be provided at all [69]. For example, a possible response to a prompt for rewriting a sentence in a particular style could be "Sounds like you are a great writer!" or "Here are more writing tips and tricks.". Such responses fail to address the specific request, offering generic or irrelevant information instead of the desired rewrite.

LLMs are prone to producing hallucinations i.e. generating content that is nonsensical or unfaithful to the provided source content. The generated outputs are grammatically correct, yet factually unsupported. Producing hallucinations could be a result of various factors [23]. While for some tasks such as creative writing, hallucinations may be beneficial, there are tasks such as summarization for which it would be disastrous [10]. For text style transfer, LLMs could occasionally misinterpret instructions in a prompt and produce outputs unrelated to the input, especially with longer prompts when it becomes difficult to distinguish between the instruction and the actual content [80].

Different methods demonstrated superiority across different evaluation aspects, highlighting the inherent trade-offs in text style transfer. Approaches that excel in content preservation tend to exhibit weaker style transfer strength, while those that perform better in style transfer strength often show worse content preservation capabilities [69], [94]. This balance remains a persistent challenge in both pre-LLM and LLM-based text style transfer which emphasizes the need for techniques that can effectively optimize both objectives.

VI. IMPLICATIONS OF LLM-BASED TEXT STYLE TRANSFER

Knowledge augmentation approaches showed promising results in several natural language generation tasks that are mostly knowledge-intensive. However, their application for enhancing text style transfer methods remains relatively unexplored. Some of the pre-LLM approaches relied on style markers of the retrieved most similar sentence [4], [5] to guide the generation process and sentiment memories [40] to learn and store information for the target sentiment, while few of the LLM-based approaches utilized sample sentences in the target style [13], [93]. These KA-based approaches hold the potential to further facilitate TST tasks and could provide valuable insights for future research. Incorporating external and/or additional knowledge could enhance the models to adapt to more diverse style transfer tasks and provide better contextual understanding.

Vast amounts of data, used for training LLMs, may include implicit societal biases, stereotypes, hallucinations, etc. Their usage for text style transfer, as well as other natural language generation tasks, could unintentionally replicate or amplify them in the generated style transferred sentence. Further training student models on such generated data could inherit those characteristics [78]. A potential misuse particularly for text style transfer is generating a sentence in the opposite direction of the task [92]. For example, making appropriate text inappropriate, transferring non-offensive text to offensive, transforming neutral sentence to toxic, or in the case of arbitrary styles making a sentence "more racist" [10]. Strong directional bias was observed for sentiment transfer where the LLM performed better in transforming a positive sentence into a negative one than the opposite [11]. Similar behavior was observed for politeness transfer with better performance for the direction from impolite to polite than the opposite [89].

Apart from standard TST tasks, LLMs were evaluated on other tasks that involve or rely on text style transfer. One application was generating textual explanations [102] in addition to generating a sentence in the target style to produce clarifications or rationales for specific transformations. This approach was used to generate synthetic explainable TST datasets for formality transfer and subjective bias correction. Student models fine-tuned on these datasets outperform generalist teacher models on the one-shot explainable text style transfer task. Another application was to modify the style of an input conversation [103], [104] or to maintain a consistent tone across exchanges, which can be critical in conversational systems. The models were able to better match the target style by incorporating multi-turn context. Additionally, text style transfer techniques were utilized to generate questions by combining a predefined style template with the internal context of a conversation [105]. This approach outperformed strong baselines in terms of diversity.

VII. CONCLUSION

This paper presented an overview of recent advancements in text style transfer with a particular focus on using LLMs, with the aim to identify research gaps and suggest further improvements. We began the discussion with a summary of the traditional pre-LLM approaches and then continued with the LLM-based approaches for text style transfer. The methods were discussed and grouped into three main categories. We hope that this overview serves as a valuable resource for future research in text style transfer that is built upon LLMs.

APPENDIX
LLM-BASED APPROACHES FOR TEXT GENERATION

Large Language Models (LLMs) demonstrated remarkable abilities in a variety of natural language processing (NLP) tasks. LLMs are constructed in accordance with the Transformer [53] architecture. These models leverage pre-training on a vast collection of datasets that enables learning the semantics and patterns inherent in language. The pre-training phase involves predicting the next word for a particular sequence [37], [52], or masked language modeling and next sentence prediction [106]. Pre-trained models could be fine-tuned for downstream tasks using task-specific datasets typically smaller than the ones used in the pre-training phase. The fine-tuning phase enables the model to learn language patterns and semantics that are specific to the task. Given that increasing the model or data size enhances the performances on downstream tasks, numerous LLMs have been proposed in the past few years: T5 [81], FLAN-T5 [95], PaLM [107], GPT-3 [62], GPT-4 [85], Falcon [70], LaMDA [63], LLaMA [71], LLaMA-2 [67], LlaMA-3 [90], Mistral [72], Gemini [108], and many others. The following subsections provide a brief overview of various approaches, including prompting techniques, fine-tuning and reinforcement learning, retrieval-augmented generation, and knowledge augmentation.

A. PROMPTING TECHNIQUES

Prompting techniques for LLMs gained an extensive research interest since GPT-3 [62] introduced in-context-learning i.e. learning from a few or no samples. Choosing the right prompt and the right number of demonstrated examples has a big impact on the performances [109], [110], [111] making prompt engineering a field of significant research attention [112], [113]. For **zero-shot prompting** [54], no examples of the task are given in the prompt, while for **one-shot** and **few-shot prompting** [55] one or *n* examples are provided as part of the prompt, respectively. **Chain-of-Thought (CoT) prompting** [56] is a new direction that has been explored recently to improve the performance on reasoning tasks. The prompt is formulated by breaking down the problem into manageable steps in a similar way that a human would approach solving a reasoning problem. **Few-shot CoT** requires manually crafted examples as part of the prompt, while **Zero-shot CoT** [114] adds *"Let's think step by step"* to the prompt before each answer. **Plan-and-Solve (PS) Prompting** [115] aims to address challenges such as missing reasoning steps by first planning subtasks and then executing them. **Skeleton-of-Thought (SoT)** [116], improves the inference time by generating key "skeleton points" that are processed in parallel. **Chain-of-Verification (CoVe)** [117] was developed with the aim of reducing hallucinations by separating the generation process into four steps: generating a baseline response, creating verification questions, answering these questions to detect hallucinations, and then producing a final verified response.

B. FINE-TUNING AND REINFORCEMENT LEARNING

The standard approach to fine-tuning LLMs is to adjust the model parameters by further training on a task-specific dataset that is usually much smaller than the datasets used for pre-training. This approach enhances the performance of the model but can be computationally expensive and resource-intensive due to the need to adjust all parameters during training. **Parameter Efficient Fine-tuning (PEFT)** [57], [118], [119] emerged as an effective solution for reducing the number of parameters for fine-tuning and memory usage while preserving performance levels similar to full fine-tuning. PEFT updates a limited set of additional parameters or modifies a subset of the pre-trained ones, thus enabling the model to retain the previously learned knowledge and at the same time, adapt to the particular task. **Prefix Tuning** [58] keeps LLM parameters frozen and optimizes a sequence of continuous task-specific vectors called prefixes. This approach obtained comparable performances to the full fine-tuning by modifying only a small subset of the parameters.

Most of the available LLMs have been trained with the objective of predicting the next token, which differs from the objective "follow the user's instructions helpfully and safely" [37], [62], [63], [120], [121]. **Instruction fine-tuning** was developed to align these two objectives by fine-tuning LLMs on pairs of human instruction for the task and desired output [95], [122]. This enables the models to respond better to instructions and reduces the need for few-shot exemplars [54], [123], [124]. **Reinforcement Learning from Human Feedback (RLHF)** [125], [126] incorporates human preferences in the training process. It uses a reward model that is trained to score the outputs according to their alignment preferences given by humans to create a dataset that is then used to further fine-tune the LLM using reinforcement learning. **Direct Preference Optimization (DPO)** [127] was developed to directly optimize an LLM to align with human preferences without explicit reinforcement learning or reward modeling. It implicitly optimizes the same objective as RLHF but is simpler to train. DPO applies a change of variable to directly define the preference loss based on the policy. With a dataset of human preferences for LLM responses, DPO optimizes the policy using a binary cross-entropy objective resulting in an optimal policy that aligns with an implicit reward function derived from the preference data.

C. RETRIEVAL AUGMENTED GENERATION

Retrieval Augmented Generation (RAG) [59] enhances the LLM performance by incorporating external knowledge in the generation process allowing access to current information and specialized knowledge that might not be included in the training data. It enables the LLMs to reference verified information, improving the reliability of their outputs and their factual accuracy. RAG approaches are generally categorized into three groups [128]: Naïve RAG, Advanced RAG, and Modular RAG. **Naïve RAG** involves basic retrieval techniques where relevant documents are retrieved to facilitate the generation process. This approach provides the basis for understanding more sophisticated RAG systems [129]. **Advanced RAG** includes pre-retrieval and post-retrieval techniques to enhance retrieval quality. Pre-retrieval is focused on optimizing the indexing structure and the original query [130], [131], [132], while post-retrieval is focused on re-ranking the retrieved information and compressing the context to be processed. **Modular RAG** introduces specialized components to enhance retrieval and processing, or reconfiguration/substitution of existing modules. Some of the approaches include a search module that adapts to various data sources using LLM-generated code [133], RAGFusion for multi-query searches to uncover diverse knowledge [134], a routing module to select the best data source [135], hybrid retrieval strategies that combine keyword, semantic and vector searches, and others.

D. KNOWLEDGE AUGMENTATION

Knowledge augmentation (KA) has been explored for LLMs in two directions [136]: querying LLMs as knowledge bases (KBs) and augmenting LLMs with knowledge. Querying LLMs as KBs aims to retrieve relevant knowledge learned from the LLM. This task was explored in various cases including casting the knowledge contained within language models into a knowledge graph [137], fine-tuning to answer questions without access to any external context or knowledge [138], or using LLMs as an additional component to create synthetic training data for student models [139]. Many NLP tasks are knowledge-intensive i.e. they require access to external knowledge sources. Several language models have been proposed for these tasks [60], [61], [140], [141].

ERNIE [60] integrates lexical, syntactic, and knowledge-based information, trained on large-scale text corpora and knowledge graphs. It was designed for tasks requiring extensive knowledge, such as relation extraction. **KnowBERT** [140] builds on BERT [106] by incorporating knowledge bases like WordNet and a subset of Wikipedia, enabling it to handle tasks like entity typing and word sense disambiguation. **LinkBERT** [141] represents text as a graph of linked documents and uses document links, such as Wikipedia hyperlinks and citation links in the biomedical domain, to enhance pre-training. **DRAGON** [61] fuses text with knowledge graph subgraphs, using masked language modeling and knowledge graph link prediction as its core pre-training tasks. It is designed to handle tasks that require reasoning about both language and knowledge.

REFERENCES

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2014, pp. 3104–3112.
[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[3] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992.
[4] J. Li, R. Jia, H. He, and P. Liang, "Delete, retrieve, generate: A simple approach to sentiment and style transfer," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 1865–1874.
[5] A. Sudhakar, B. Upadhyay, and A. Maheswaran, "'Transforming delete, retrieve, generate approach for controlled text style transfer," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3260–3270.
[6] J. Lee, "Stable style transformer: Delete and generate approach with encoder–decoder for text style transfer," 2020, *arXiv:2005.12086*.
[7] Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan, "Style transfer in text: Exploration and evaluation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, vol. 32, Apr. 2018, pp. 663–670.
[8] Y. Zhao, W. Bi, D. Cai, X. Liu, K. Tu, and S. Shi, "Language style transfer from sentences with arbitrary unknown styles," 2018, *arXiv:1808.04071*.
[9] F. Luo, P. Li, J. Zhou, P. Yang, B. Chang, X. Sun, and Z. Sui, "A dual reinforcement learning framework for unsupervised text style transfer," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 5116–5122.
[10] E. Reif, D. Ippolito, A. Yuan, A. Coenen, C. Callison-Burch, and J. Wei, "A recipe for arbitrary text style transfer with large language models," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 837–848.
[11] M. Suzgun, L. Melas-Kyriazi, and D. Jurafsky, "Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 2195–2222.
[12] S. Mukherjee and O. Dusek, "Leveraging low-resource parallel data for text style transfer," in *Proc. 16th Int. Natural Lang. Gener. Conf.*, 2023, pp. 388–395.
[13] C. Zong, Y. Chen, W. Lu, J. Shao, and Y. Zhuang, "ProSwitch: Knowledge-guided instruction tuning to switch between professional and non-professional responses," 2024, *arXiv:2403.09131*.
[14] S. Mukherjee and O. Dušek, "Text style transfer: An introductory overview," 2024, *arXiv:2407.14822*.
[15] S. Mukherjee, M. Lango, Z. Kasner, and O. Dušek, "A survey of text style transfer: Applications and ethical implications," 2024, *arXiv:2407.16737*.

[16] E. Troiano, A. Velutharambath, and R. Klinger, "From theories on styles to their transfer in text: Bridging the gap with a hierarchical survey," *Natural Lang. Eng.*, vol. 29, no. 4, pp. 849–908, Jul. 2023.

[17] D. Jin, Z. Jin, Z. Hu, O. Vechtomova, and R. Mihalcea, "Deep learning for text style transfer: A survey," *Comput. Linguistics*, vol. 48, no. 1, pp. 155–205, Apr. 2022.

[18] M. Toshevska and S. Gievska, "A review of text style transfer using deep learning," *IEEE Trans. Artif. Intell.*, vol. 3, no. 5, pp. 669–684, Oct. 2022.

[19] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, and S. Mirjalili, "Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects," *Authorea Preprints*, vol. 2023, pp. 1–26, Nov. 2023.

[20] W. Xin Zhao et al., "A survey of large language models," 2023, *arXiv:2303.18223*.

[21] Z. Guo, R. Jin, C. Liu, Y. Huang, D. Shi, Supryadi, L. Yu, Y. Liu, J. Li, B. Xiong, and D. Xiong, "Evaluating large language models: A comprehensive survey," 2023, *arXiv:2310.19736*.

[22] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2024, *arXiv:2402.06196*.

[23] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," 2023, *arXiv:2311.05232*.

[24] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, and J. Han, "Large language models on graphs: A comprehensive survey," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 12, pp. 8622–8642, Dec. 2024.

[25] S. Rao and J. Tetreault, "Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 129–140.

[26] W. Xu, A. Ritter, B. Dolan, R. Grishman, and C. Cherry, "Paraphrasing for style," in *Proc. COLING*, Jan. 2012, pp. 2899–2914.

[27] W. Xu, "Data-driven approaches for paraphrasing across language variations," Ph.D. dissertation, Dept. Comput. Sci., New York Univ., New York, NY, USA, 2014.

[28] R. Pryzant, R. D. Martinez, N. Dass, S. Kurohashi, D. Jurafsky, and D. Yang, "Automatically neutralizing subjective bias in text," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 480–489.

[29] V. Logacheva, D. Dementieva, S. Ustyantsev, D. Moskovskiy, D. Dale, I. Krotova, N. Semenov, and A. Panchenko, "ParaDetox: Detoxification with parallel data," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 6804–6818.

[30] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 193–202.

[31] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 507–517.

[32] A. Madaan, A. Setlur, T. Parekh, B. Poczos, G. Neubig, Y. Yang, R. Salakhutdinov, A. W. Black, and S. Prabhumoye, "Politeness transfer: A tag and generate approach," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., 2020, pp. 1869–1881.

[33] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.

[34] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc. acl workshop intrinsic extrinsic Eval. measures Mach. Transl. and/or summarization*, Jun. 2005, pp. 65–72.

[35] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics (ACL), 2004, pp. 74–81.

[36] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2019.

[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[38] Q. Liu, J. Qin, W. Ye, H. Mou, Y. He, and K. Wang, "Adaptive prompt routing for arbitrary text style transfer with pre-trained language models," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, vol. 38, no. 17, pp. 18689–18697.

[39] G. Luo, Y. Han, L. Mou, and M. Firdaus, "Prompt-based editing for text style transfer," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2023, pp. 5740–5750.

[40] Y. Zhang, J. Xu, P. Yang, and X. Sun, "Learning sentiment memories for sentiment modification without parallel data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1103–1108.

[41] G. Lample, S. Subramanian, E. M. Smith, L. Denoyer, M. Ranzato, and Y.-L. Boureau, "Multiple-attribute text rewriting," in *Proc. 7th Int. Conf. Learn. Represent.*, New Orleans, LA, USA, Sep. 2018. [Online]. Available: https://openreview.net/forum?id=H1g2NhC5KQ

[42] N. Dai, J. Liang, X. Qiu, and X.-J. Huang, "Style Transformer: Unpaired text style transfer without disentangled latent representation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5997–6007.

[43] R. Xu, T. Ge, and F. Wei, "Formality style transfer with hybrid textual annotations," 2019, *arXiv:1903.06353*.

[44] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS Workshop Deep Learn.*, Dec. 2014.

[45] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.

[46] T. Shen, T. Leí, R. Barzilay, and T. Jaakkola, "Style transfer from non-parallel text by cross-alignment," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, May 2017, pp. 6830–6841.

[47] X. Yi, Z. Liu, W. Li, and M. Sun, "Text style transfer via learning style instance supported latent space," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 3801–3807.

[48] J. Xu, X. Sun, Q. Zeng, X. Zhang, X. Ren, H. Wang, and W. Li, "Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 979–988.

[49] H. Gong, S. Bhat, L. Wu, J. Xiong, and W.-M. Hwu, "Reinforcement learning based text style transfer without parallel training corpus," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 3168–3180.

[50] B. Upadhyay, A. Sudhakar, and A. Maheswaran, "Efficient reinforcement learning for unsupervised controlled text generation," 2022, *arXiv:2204.07696*.

[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[52] A. Radford et al., "Improving language understanding by generative pre-training," OpenAI Blog, 2018.

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[54] J. Lee, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2021.

[55] T. Schick and H. Schütze, "Few-shot text generation with natural language instructions," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 390–402.

[56] J. Lee, X. Wang, D. Schuurmans, M. Bosma, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Jan. 2022, pp. 24824–24837.

[57] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. D. Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proc. 36th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds., Jun. 2019, pp. 2790–2799.

[58] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4582–4597.

[59] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 9459–9474.

[60] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1441–1451.

[61] M. Yasunaga, A. Bosselut, H. Ren, X. Zhang, C. D. Manning, P. Liang, and J. Leskovec, "Deep bidirectional language-knowledge graph pretraining," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 37309–37323.

[62] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., Jan. 2020, pp. 1877–1901.

[63] R. Thoppilan et al., "LaMDA: Language models for dialog applications," 2022, *arXiv:2201.08239*.

[64] B. Wang. (May 2021). *Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model With JAX*. [Online]. Available: https://github.com/kingoflolz/mesh-transformer-jax

[65] S. Black, G. Leo, P. Wang, C. Leahy, and S. Biderman. (2021). *GPT-Neo: Large Scale Autoregressive Language Modeling With Mesh-Tensorflow*, doi: 10.5281/zenodo.5297715.

[66] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019, *arXiv:1910.13461*.

[67] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.

[68] OpenAI. (2021). *ChatGPT*. [Online]. Available: https://openai.com/

[69] S. Mukherjee, A. K. Ojha, and O. Dušek, "Are large language models actually good at text style transfer?" 2024, *arXiv:2406.05885*.

[70] E. Almazrouei et al., "Falcon-40b: An open large language model with state-of-the-art performance," in *Proc. Findings Assoc. Comput. Linguistics, ACL*, 2023, pp. 10755–10773.

[71] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[72] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7B," 2023, *arXiv:2310.06825*.

[73] T. Le Scao et al., "BLOOM: A 176B-parameter open-access multilingual language model," 2023, *arXiv:2211.05100*.

[74] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang, "GLM: General language model pretraining with autoregressive blank infilling," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 320–335.

[75] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "OPT: Open pre-trained transformer language models," 2022, *arXiv:2205.01068*.

[76] L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, N. Sarrazin, O. Sanseviero, A. M. Rush, and T. Wolf, "Zephyr: Direct distillation of LM alignment," 2023, *arXiv:2310.16944*.

[77] P. Liu, L. Wu, L. Wang, S. Guo, and Y. Liu, "Step-by-step: Controlling arbitrary style in text with large language models," in *Proc. Joint Int. Conf. Comput. Linguistics, Lang. Resour. Eval. (LREC-COLING)*, 2024, pp. 15285–15295.

[78] C. Zhang, H. Cai, Y. Li, Y. Wu, L. Hou, and M. Abdul-Mageed, "Distilling text style transfer with self-explanation from LLMs," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2024, pp. 200–211.

[79] R. Anil et al., "PaLM 2 technical report," 2023, *arXiv:2305.10403*.

[80] L. Pan, Y. Lan, Y. Li, and W. Qian, "Unsupervised text style transfer via LLMs and attention masking with multi-way interactions," 2024, *arXiv:2402.13647*.

[81] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[82] A. Zarcone and F. Kopf, "Bubble up—A fine-tuning approach for style transfer to community-specific subreddit language," in *Proc. 3rd Workshop Comput. Linguistics Political Social Sci.*, 2023, pp. 46–58.

[83] M. La Quatra, G. Gallipoli, and L. Cagliero, "Self-supervised text style transfer using cycle-consistent adversarial networks," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 5, pp. 1–38, Oct. 2024.

[84] A. Bhandarkar, R. Wilson, A. Swarup, and D. Woodard, "Emulating author style: A feasibility study of prompt-enabled text stylization with off-the-shelf LLMs," in *Proc. 1st Workshop Personalization Generative AI Syst.*, 2024, pp. 76–82.

[85] OpenAI, "GPT-4 technical report," 2023, *arXiv:2303.08774*.

[86] H. Mai, W. Jiang, and Z.-H. Deng, "Prefix-tuning based unsupervised text style transfer," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2023, pp. 14847–14856.

[87] K. Konen, S. Jentzsch, D. Diallo, P. Schütt, O. Bensch, R. E. Baff, D. Opitz, and T. Hecking, "Style vectors for steering generative large language models," in *Proc. Findings Assoc. Comput. Linguistics, EACL*, 2024, pp. 782–802.

[88] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto. (2023). *Stanford Alpaca: An Instruction-Following LLaMA Model*. [Online]. Available: https://github.com/tatsu-lab/stanford_alpaca

[89] W. Lai, V. Hangya, and A. Fraser, "Style-specific neurons for steering LLMs in text style transfer," 2024, *arXiv:2410.00593*.

[90] A. Dubey et al., "The Llama 3 herd of models," 2024, *arXiv:2407.21783*.

[91] M. Deng, J. Wang, C.-P. Hsieh, Y. Wang, H. Guo, T. Shu, M. Song, E. Xing, and Z. Hu, "RLPrompt: Optimizing discrete text prompts with reinforcement learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 3369–3391.

[92] T. Ziegenbein, G. Skitalinskaya, A. Bayat Makou, and H. Wachsmuth, "LLM-based rewriting of inappropriate argumentation using reinforcement learning from machine feedback," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics*, Bangkok, Thailand, 2024, pp. 4455–4476.

[93] M. Toshevska and S. Gievska, "Large language models for text style transfer: Exploratory analysis of prompting and knowledge augmentation techniques," in *Intell. Environments, Combined Proc. Workshops Demos Videos Session*, Jun. 2024, pp. 134–142.

[94] M. Toshevska, S. Kalajdziski, and S. Gievska, "Style knowledge graph: Augmenting text style transfer with knowledge graphs," in *Proc. Int. Workshop Generative AI Knowl. Graphs@ GenAIK-COLING*, 2025, pp. 123–135.

[95] H. Won Chung et al., "Scaling instruction-finetuned language models," 2022, *arXiv:2210.11416*.

[96] X. Liang, H. Wang, S. Song, M. Hu, X. Wang, Z. Li, F. Xiong, and B. Tang, "Controlled text generation for large language model with dynamic attribute graphs," 2024, *arXiv:2402.11218*.

[97] M. Javaheripi et al., "Phi-2: The surprising power of small language models," *Microsoft Res. Blog*, vol. 1, no. 3, p. 3, 2023.

[98] S. J. Russell and P. Norvig, *Artificial Intelligence a Modern Approach*. London, U.K.: Pearson, 2010.

[99] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2223–2232.

[100] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[101] T. Ziegenbein, S. Syed, F. Lange, M. Potthast, and H. Wachsmuth, "Modeling appropriate language in argumentation," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 4344–4363.

[102] A. Saakyan and S. Muresan, "ICLEF: In-context learning with expert feedback for explainable style transfer," 2023, *arXiv:2309.08583*.

[103] S. Roy, R. Shu, N. Pappas, E. Mansimov, Y. Zhang, S. Mansour, and D. Roth, "Conversation style transfer using few-shot learning," in *Proc. 13th Int. Joint Conf. Natural Lang. Process. 3rd Conf. Asia–Pacific Chapter Assoc. Comput. Linguistics*, 2023, pp. 119–143.

[104] J. Chen, "LMStyle benchmark: Evaluating text style transfer for chatbots," 2024, *arXiv:2403.08943*.

[105] Q. Gou, Z. Xia, B. Yu, H. Yu, F. Huang, Y. Li, and N. Cam-Tu, "Diversify question generation with retrieval-augmented style transfer," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 1677–1690.

[106] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jan. 2018, pp. 4171–4186.

[107] A. Chowdhery et al., "PaLM: Scaling language modeling with pathways," *J. Mach. Learn. Res.*, vol. 24, no. 240, pp. 1–113, Jan. 2022.

[108] G. Team et al., "Gemini: A family of highly capable multimodal models," 2023, *arXiv:2312.11805*.

[109] L. Reynolds and K. McDonell, "Prompt programming for large language models: Beyond the few-shot paradigm," in *Proc. Extended Abstr. CHI Conf. Hum. Factors Comput. Syst.*, May 2021, pp. 1–7.

[110] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 3816–3830.

[111] D. Zhou, N. Schärli, L. Hou, J. Lee, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, and Q. V. Le, "Least-to-most prompting enables complex reasoning in large language models," in *Proc. 11th Int. Conf. Learn. Represent.*, Jan. 2022.

[112] P. Sahoo, A. Kumar Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," 2024, *arXiv:2402.07927*.

[113] S. Vatsal and H. Dubey, "A survey of prompt engineering methods in large language models for different NLP tasks," 2024, *arXiv:2407.12994*.

[114] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 22199–22213.

[115] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 2609–2634.

[116] X. Ning, Z. Lin, Z. Zhou, Z. Wang, H. Yang, and Y. Wang, "Skeleton-of-thought: Prompting LLMs for efficient parallel generation," in *Proc. 12th Int. Conf. Learn. Represent.*, 2024, pp. 1–51.

[117] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. E. Weston, "Chain-of-verification reduces hallucination in large language models," in *Findings of the Association for Computational Linguistics (ACL)*, 2024, pp. 3563–3578.

[118] L. Xu, H. Xie, S.-Z. Joe Qin, X. Tao, and F. Lee Wang, "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment," 2023, *arXiv:2312.12148*.

[119] Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang, "Parameter-efficient fine-tuning for large models: A comprehensive survey," 2024, *arXiv:2403.14608*.

[120] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res.*, vol. 23, no. 120, pp. 1–39, Jan. 2021.

[121] J. W. Rae et al., "Scaling language models: Methods, analysis & insights from training gopher," 2021, *arXiv:2112.11446*.

[122] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang, "Instruction tuning for large language models: A survey," 2023, *arXiv:2308.10792*.

[123] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. E. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Jan. 2022, pp. 27730–27744.

[124] V. Sanh et al., "Multitask prompted training enables zero-shot task generalization," in *Proc. 10th Int. Conf. Learn. Represent.*, Jan. 2021.

[125] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jan. 2017, pp. 4302–4310.

[126] T. Kaufmann, P. Weng, V. Bengs, and E. Hüllermeier, "A survey of reinforcement learning from human feedback," 2023, *arXiv:2312.14925*.

[127] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2024, pp. 53728–53741.

[128] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," 2023, *arXiv:2312.10997*.

[129] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting in retrieval-augmented large language models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 5303–5315.

[130] W. Peng, G. Li, Y. Jiang, Z. Wang, D. Ou, X. Zeng, D. Xu, T. Xu, and E. Chen, "Large language model based long-tail query rewriting in taobao search," in *Proc. Companion ACM Web Conf.*, May 2024, pp. 20–28.

[131] H. Zheng, S. Mishra, X. Chen, H.-T. Cheng, Q. V. Le, and D. Zhou, "Take a step back: Evoking reasoning via abstraction in large language models," in *Proc. 12th Int. Conf. Learn. Represent.*, Jan. 2023.

[132] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 1762–1777.

[133] X. Wang, Q. Yang, Y. Qiu, J. Liang, Q. He, Z. Gu, Y. Xiao, and W. Wang, "KnowledGPT: Enhancing large language models with retrieval and storage access on knowledge bases," 2023, *arXiv:2308.11761*.

[134] Z. Rackauckas, "Rag-fusion: A new take on retrieval augmented generation," *Int. J. Natural Lang. Comput.*, vol. 13, no. 1, pp. 37–47, Feb. 2024.

[135] X. Li, E. Nie, and S. Liang, "From classification to generation: Insights into crosslingual retrieval augmented ICL," in *Proc. NeurIPS Workshop Instruct. Tuning Instruct. Following*, Jan. 2023.

[136] C. Zhen, Y. Shang, X. Liu, Y. Li, Y. Chen, and D. Zhang, "A survey on knowledge-enhanced pre-trained language models," 2022, *arXiv:2212.13428*.

[137] C. Wang, X. Liu, and D. Song, "Language models are open knowledge graphs," 2020, *arXiv:2010.11967*.

[138] A. Roberts, C. Raffel, and N. Shazeer, "How much knowledge can you pack into the parameters of a language model?" in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 5418–5426.

[139] J. Chung, E. Kamar, and S. Amershi, "Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 575–593.

[140] M. E. Peters, M. Neumann, R. Logan, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith, "Knowledge enhanced contextual word representations," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 43–54.

[141] M. Yasunaga, J. Leskovec, and P. Liang, "LinkBERT: Pretraining language models with document links," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 8003–8016.

MARTINA TOSHEVSKA (Graduate Student Member, IEEE) received the B.S. and M.S. degrees from Ss. Cyril and Methodius University in Skopje, North Macedonia, in 2018 and 2019, respectively, where she is currently pursuing the Ph.D. degree. She is also a Research and Teaching Assistant of computer science with the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje. Her research interests include natural language processing, graph analysis, deep learning, artificial intelligence, and their applications.

SONJA GIEVSKA received the B.S. degree from Ss. Cyril and Methodius University in Skopje, North Macedonia, in 1986, the M.S. degree from the University of Zagreb, Croatia, in 1996, and the Ph.D. degree from The George Washington University, Washington, DC, USA, in 2004. She is currently a Professor of computer science with the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje. Her research interests include machine learning, artificial intelligence, and their applications.