A Survey of Controllable Text Generation Using Transformer-based Pre-trained Language Models

HANQING ZHANG, HAOLIN SONG, and SHAOYU LI, Beijing Institute of Technology, China
MING ZHOU, Langboat Technology, China
DAWEI SONG, Beijing Institute of Technology, China

Controllable Text Generation (CTG) is an emerging area in the field of natural language generation (NLG). It is regarded as crucial for the development of advanced text generation technologies that better meet the specific constraints in practical applications. In recent years, methods using large-scale pre-trained language models (PLMs), in particular the widely used Transformer-based PLMs, have become a new paradigm of NLG, allowing generation of more diverse and fluent text. However, due to the limited level of interpretability of deep neural networks, the controllability of these methods needs to be guaranteed. To this end, controllable text generation using Transformer-based PLMs has become a rapidly growing yet challenging new research hotspot. A diverse range of approaches have emerged in the past 3 to 4 years, targeting different CTG tasks that require different types of controlled constraints. In this article, we present a systematic critical review on the common tasks, main approaches, and evaluation methods in this area. Finally, we discuss the challenges that the field is facing, and put forward various promising future directions. To the best of our knowledge, this is the first survey article to summarize the state-of-the-art CTG techniques from the perspective of Transformer-based PLMs. We hope it can help researchers and practitioners in the related fields to quickly track the academic and technological frontier, providing them with a landscape of the area and a roadmap for future research.

CCS Concepts: • **Computing methodologies → Natural language processing;**

Additional Key Words and Phrases: Controllable text generation, pre-trained language models, Transformer, controllability, systematic review

ACM Reference format:
Hanqing Zhang, HaoLin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A Survey of Controllable Text Generation Using Transformer-based Pre-trained Language Models. *ACM Comput. Surv.* 56, 3, Article 64 (October 2023), 37 pages.
https://doi.org/10.1145/3617680

D. Song is also with The Open University, UK.

This work is supported in part by Natural Science Foundation of Beijing (grant no. 4222036).

Authors' addresses: H. Zhang, H. Song, S. Li, and D. Song (Corresponding author), Beijing Institute of Technology, No. 5 South Street, Zhongguancun, Haidian District, Beijing, 100081, China; e-mails: {zhanghanqing, hlsong}@bit.edu.cn, lishaoyuxl@foxmail.com; M. Zhou, Langboat Technology, No. 52 Beisihuan West Road, Haidian District, Beijing, 100081, China; e-mail: zhouming@chuangxin.com.

0360-0300/2023/10-ART64 $15.00
https://doi.org/10.1145/3617680

1 INTRODUCTION

Natural language generation (NLG) is regarded as complementary to natural-language understanding (NLU), an essential branch of natural language processing (NLP). Contrary to the task of NLU, which aims to disambiguate an input text to produce a single normalized representation of the ideas expressed in the text, NLG mainly focuses on transforming the potential representations into specific, self-consistent natural language text [47]. In other words, NLU aims to develop an intelligent machine that can read and understand human language, while NLG enables computers to write like humans. As an embodiment of advanced artificial intelligence, NLG technologies play a crucial role in a range of applications, such as dialogue systems, advertising, marketing, story generation, and data augmentation.

Making text generation controllable is an important and fundamental issue in NLG. Some concrete examples are shown in Figure 1. Generally speaking, an NLG system should be able to reliably generate texts that meet certain controllable constraints that are imposed by the targeted applications and users. In general, these constraints are task specific. For example, the task of story generation always needs to control the storyline and the ending. In the task of dialogue response generation, controlling the emotion [67], persona [160] and politeness, etc., is often required. For generation-based data augmentation [42], it is necessary to ensure the data distribution balance in different domains. Moreover, for ethical development [6] of AI applications, it is crucial to avoid generating mindless and offensive content such as gender bias, racial discrimination, and toxic words. Therefore, the controllability of an NLG system is crucial for it to generate significant practical value in real applications.

In recent years, the development of deep learning (DL) has given rise to a series of studies on DL-driven controllable text generation (CTG), which has brought genuine breakthroughs in this field. Early approaches are based on sequential models and style embedding [34, 65], which have achieved some promising progress. After that, there is a surge of methods based on deep generative models, such as Variational Autoencoders (VAEs) [48, 125, 138, 142, 149, 154], Generative Adversarial Networks (GANs) [117, 140], and Energy-based Models [8, 25, 135, 166]. Deep learning–based methods are capable of an end-to-end learning in a data-driven way to learn low-dimensional dense vectors that implicitly represent the linguistic features of text. Such representation is also useful to avoid the bias of hand-crafted features and has shown great potential in text generation.

However, the success of the above DL-based methods relies heavily on large-scale datasets, posing a challenge for supervised and cross-domain text generation tasks. Since 2018, large-scale pre-trained language models (PLMs) such as BERT [27], RoBERTa [82], GPT [107], T5 [108], and mBART [80] have gradually become a new paradigm of NLP. Owing to its use of large corpora and unsupervised learning based on the Transformer structure, PLMs are believed to have learned a great deal of semantic and syntactic knowledge from the data, and only fine-tuning is required for downstream tasks to get the state-of-the-art (SOTA) performance. In terms of NLG, PLMs have learned from a large number of corpus materials to model the distribution of natural language to a large extent so that they are able to generate texts of unprecedented quality [25]. Moreover, a large-scale PLM itself can be viewed as a well-informed knowledge base, making it possible to generate text without the need for external domain knowledge. Nevertheless, PLMs are neural network based, which essentially are still black boxes, lacking a good level of interpretability. Those models always generate texts according to the latent representation of the context. Thus, it is difficult to control them to generate content that humans want (i.e., controllability issues). How to improve the interpretability and controllability of the PLM-based methods for generating text has become a hot research topic.

In the above application and research contexts, PLM-based methods are becoming the mainstream of CTG research and are expected to bring milestone progress. As a rapidly growing yet

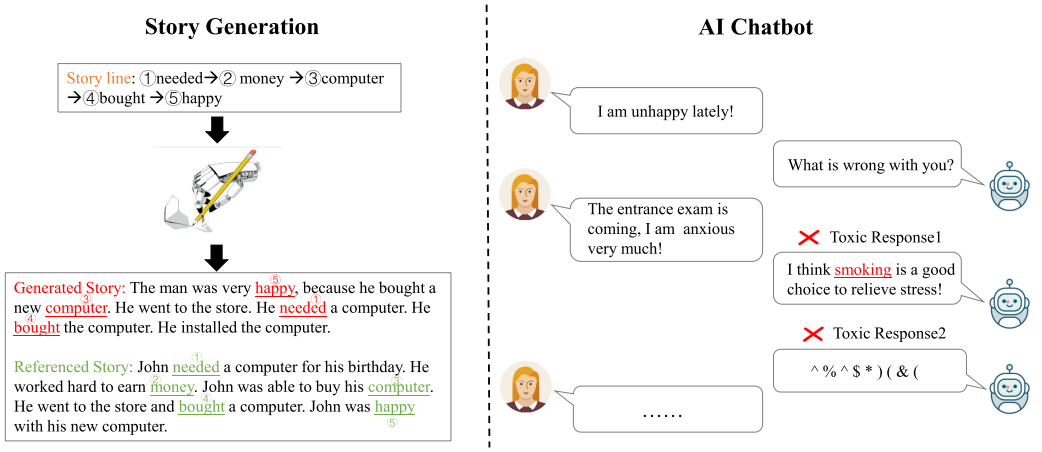


Fig. 1. Toy examples for controllable text generation. The left-hand side shows an application of story generation, which needs to ensure that the generated story matches the key elements provided by the storyline and the order in which they appear. The right-hand side shows an application of dialogue text generation. One important controlled requirement is to avoid generating toxic responses, such as the harmful introductory advice about "smoking" and gibberish as shown in the figure.

challenging research field, there is an urgent need for a comprehensive critical review of the current literature to draw a landscape of the area and set out a roadmap for promising future directions. There are some existing surveys on CTG [100], but they lack (1) a systematic review of representative application tasks, main approaches, and evaluation methodologies of CTG; and (2) a tracking of the latest large-scale PLM-based CTG approaches. In this article, we provide an introduction to the main tasks and evaluation metrics related to CTG, a dedicated and comprehensive literature review on CTG approaches using PLMs, and, finally, an outlook on the possible future research directions. We hope that this survey article will help researchers and practitioners to quickly capture the overall picture as well as detailed cutting-edge methods in PLM-based CTG, and promote the further development of this promising area.

The remainder of the article is organized as follows: Section 2 gives a brief introduction to the two critical aspects of the area, i.e., the fundamental concepts of CTG and PLMs. Then, we divide the main approaches to PLM-based CTG into three categories and discuss them in more detail in Section 3. Section 4 summarizes the relevant evaluation methodologies and metrics for CTG. In Section 5, we discuss the challenges that the field is facing and put forward a number of promising future directions. Finally, we conclude the article in Section 6. All the literature appearing in this article is filtered following two rules. First, we tend to select the latest papers that appeared within 3 to 4 years, ensuring the timeliness of the surveyed works. Second, we preferably select the works that are influential in the NLP community, e.g., the papers published in the leading conferences or journals in the NLP field, such as ACL, EMNLP, NAACL and TACL, and the works that are highly cited or have received widespread attention in the open source community.

2 AN INTRODUCTION TO CONTROLLABLE TEXT GENERATION AND PRE-TRAINED LANGUAGE MODELS

This article is closely related to two key aspects: controllable text generation and pre-trained language models, which will be briefly introduced in this section.

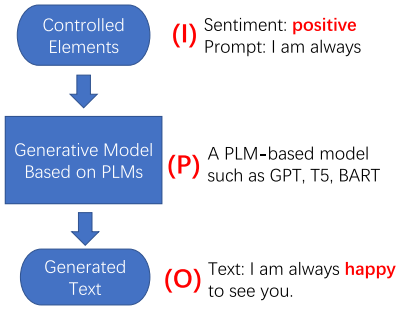


Fig. 2. The IPO of controlled text generation. A typical CTG system consists of three components: the controlled element (controlled condition and source text) as input (I), the generative model as the process (P), and the generated text satisfying the input control condition as output (O).

2.1 Controllable Text Generation

Controllable text generation (CTG) refers to the task of generating text according to the given controlled element [100]. As shown in Figure 2, a typical CTG system consists of three components: the controlled element, including a controlled condition (e.g., a positive sentiment) and a source text (which can be vacant or just a text prompt in some applications) as input (I); the generative model (e.g., a PLM-based model) as the process (P), and the generated text satisfying the input control condition, as output (O). Take sentiment control as an example. If we want to generate a sentence with a positive emotion, then the condition "positive sentiment" and corresponding prompt "I am always" are taken as the control element and input into a PLM-based generative model. The output sentence's sentiment disposition would satisfy the controlled element, such as "I am always happy to see you".

Depending on different applications, the attributes of control conditions can be in different forms and connotations. They could range from text attribution (such as sentiment, topic, and keywords); author style and speaker identity of the person writing the text (such as gender and age); text genre and formats (such as poems, couplets); ordering of events (such as storylines); to structured data description (such as table-to-text and Knowledge Graph (KG)-to-text generation). All of the above task types can be formalized mathematically in a unified form as follows.

Given a vocabulary $\mathscr{V}$, the goal of CTG is to generate a target text $Y = \{y_1, y_2, \ldots, y_n\}$, where $y_n \in \mathscr{V}$, with respect to a control element denoted as C. Then CTG can be formally described as

$$P(Y|C) = p(y_1, y_2, \ldots, y_n|C). \tag{1}$$

The specific expression of C may vary according to different tasks. We divide the commonly used control conditions into three categories, i.e., semantic, structural, and lexical constraints, as shown in Figure 3. The sentence Y generated by a CTG model is expected to satisfy the constraint conditions while conforming to the general natural language characteristics, such as fluency, rationality and readability, to the greatest extent.

2.2 Tasks and Applications Involving CTG

Controllability is the fundamental problem of text generation, which is indeed required by almost all text generation scenes. Here, we only focus on those with explicitly controlled conditions and goals. Table 1 summarizes typical tasks and applications involving CTG, with a description of the input/output, controlled aspects, and representative references. They are explained in more detail below:

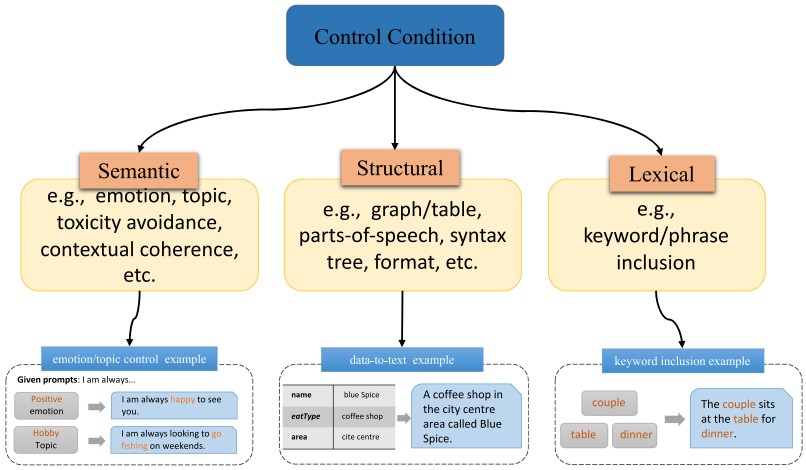


Fig. 3. The taxonomy of control conditions. We generally divide the control conditions into three types: (1) Semantic: it generally refers to content control at the semantic level, which is the reverse process of some text understanding tasks, such as emotion/topic classification and toxic detection. (2) Structural: it is opposed to the semantic control, and refers to the control over the structure level of the generated text, which is always related to text structure analysis tasks such as information extraction and sentence parsing. (3) Lexical: it represents a type of controllable text generation task at the vocabulary level, such as generating text that contains certain keywords.

Table 1. General Overview of the Tasks Involving CTG

Task	Input & Output	Controllable Aspects
Attribute-based Generation	Input: Keywords, discrete attributes Output: Attribute-specific sentence	Topic [24, 59, 133, 142], tense [83], politeness [114, 119], sentiment [18, 24, 44, 59, 115, 158, 159], keywords [10, 45, 143, 163]
Dialogue Generation	Input: dialogue content, additional structural information (e.g., persona, emotion, intent, template, etc.) Output: Dialogue response	Persona [126, 127, 157, 167, 168] politeness[36, 92], Sentiment [35, 113, 129, 145], Template[55, 139], ground-truth reference [97, 105, 109, 147]
Storytelling	Input: Story elements Output: Story paragraph	Story structure [32, 33, 40], story ending [85, 132], topic [16, 73, 141, 153], persona [15, 76]
Data to Text	Input: Table/graph data Output: The texts describing the data information	Structural information [102, 111, 112, 128, 131, 165]
Data Augmentation	Input: Original text, pre-defined slot values Output: Text that specific features are replaced	Pre-defined slot values [1, 79, 86]
Debiasing	Input: Biased text/biased model Output: Unbiased text/unbiased model	Political bias [78], gender bias [28, 104], subjective bias [101], social bias [4, 121], sentiment bias [51], toxicity [62, 74])
Format Control	Input: Desired Format, prompt text Output: Text in pre-defined formation	Format [66, 70, 120, 122]

We enumerate 7 categories of tasks involving CTG. For each category, we briefly describe the input and output, and list the representative references based on controllable aspects.

- **Attribute-based Generation:** Attribute-based CTG aims to generate natural language sentences that satisfy specific attributes, such as topic, emotion, and keywords. Precisely controlling the various attributes of sentences is an essential requirement of intelligent writing. By combining multiple control attributes, the system can, in theory, create interpretable and controllable paragraphs or articles. Thus, attribute-controlled text generation has always been the focus of attention in the field of text generation.

- **Dialogue Generation:** The goal of dialogue systems is to build an agent who can mimic human conversations using natural language. Generative dialogue models often have higher requirements in consistency, semantics, and interactiveness [50]. Therefore, the constraints on emotion, the speaker's personal style, dialogue intent/action, etc., are used to control the dialogue responses and improve the interactivity of dialogue systems.
- **Storytelling:** Storytelling requires the model to generate texts with a complete narrative logic, which needs a higher level of control on long text generation. Storylines and story endings are often regarded as controlled conditions, and the model needs to produce stories with fluent text and sound plots according to the given controlled conditions.
- **Data to Text:** The main goal of data-to-text generation is to convert non-linguistic structured data (e.g., a table or a graph) into natural language text, which can be applied in tasks such as weather forecasting, health care [11], and so on. The controllability of data-to-text tasks is to ensure that the generated text contains information manifested in the original structure data.
- **Data Augmentation:** Neural networks heavily rely on a large amount of labeled data. The importance of data augmentation is now becoming more conspicuous given the significant cost of data collection and cleaning. Since recent neural network models are capable of generating near-realistic text, it is possible to utilize them to expand existing datasets and even create new data. Identifying and replacing some entities in the given text or generating new sentences according to given attributes through CTG has become an efficient way for data augmentation.
- **Debiasing:** Biased training data may cause a model to learn incorrect knowledge, and accordingly produce biased results. Therefore, text debiasing has attracted increasing attention. Debiasing by rewriting the biased text or changing the data distribution of CTG-generated text has been shown feasible. The major controllable aspects in this task include gender, race, and toxicity.
- **Format Control Tasks:** There are also CTG tasks that need to control the format of the generated text, such as text length and rhythm. For example, the task of generating traditional Chinese poetry and couplets has strict requirements in format, including the number of words, structure, etc.

2.3 Transformer-based Pre-trained Language Models

Recent years have witnessed the emergence and successful applications of large-scale pre-trained language models (PLMs). They are regarded as a revolutionary breakthrough in deep learning and NLP. During the pre-training stage, the use of large-scale unlabeled data can provide a strong support to an increased model scale and a better grasp of the diverse knowledge (e.g., linguistic knowledge, common sense, facts, expertise, etc.) in the data. State-of-the-art (SOTA) performance has been achieved in downstream tasks by fine-tuning the PLMs based on only a small amount of supervised data.

The early work related to PLMs can be traced back to NNLM [7], word2Vector [88], and ELMo [98]. More recently, the pre-trained models based on Transformer [137], a purely attention-based deep neural network, have greatly improved the performance in almost all NLP tasks and become the mainstream. The representative PLM infrastructures are now mainly based on Transformer [137] or its variants, such as Transformer-XL [23], Longformer [5], and Reformer [61]. Thus, this article is focused on Transformer-based PLMs. The objectives of model learning mainly include masked language modeling (MLM), corrupted text reconstruction (CTR), etc. In order for a good understanding of them, we give a summary of the representative PLMs in Table 2 according to their data construction modes, model infrastructures, pre-training tasks, and main applications.

Table 2. An Overview of the Characteristics of Typical PLMs

Name	Model Type	Infrastructures	Pre-training Task	Main Application
BERT [27]	AE	Encoder	MLM+NSP	NLU
XLNET [155]	AR	Transformer-XL	PLM	NLU
GPT2 [107], GPT3 [9]	AR	Decoder	SLM	NLG
T5 [108]	Seq2Seq	Encoder+ Decoder	CTR	NLG+NLU
mBART [80]	Seq2Seq	Encoder+ Decoder	FTR	NLG
UniLM [29]	AE+AR+Seq2Seq	Encoder+Decoder	SLM+CTR+NSP	NLG+NLU
ERNIE-T [164]	AE	Encoder	CTR+NSP	NLU
XLM [21]	Seq2Seq	Encoder+ Decoder	TLM	NLG
Bloom [116]	AR	Decoder	SLM	NLG
PaLM [19]	AR	Decoder	SLM	NLG
LLaMA [134]	AR	Decoder	SLM	NLG

"MLM" means "Mask Language Model"; "NSP" means "Next Sentence Prediction"; "SLM" means "Standard Language Mode"; "CTR" means "Corrupted Text Reconstruction"; "FTR" means "Full Text Reconstruction"; "PLM" means "Permutation Language Modeling"; and "TLM" means "Translation Language Modeling". The detailed definition of the pre-training task can be seen in literature [77].

Here, we roughly divide the existing PLMs into the following three categories and provide a brief introduction to each of them.

Auto-Encoding (AE) Models: This type of PLM is constructed based on destroying the input text in some way, such as masking some words of a sentence and then trying to reconstruct the original text. Typical examples of this type include BERT, ROBERTA, and ERNIE. Because these models aim to build bidirectional encoding representations of the entire sentences, their infrastructures often correspond to the encoder part of Transformer, which does not contain any masked attention mechanism, and all input can be accessed at each location. They can then be fine-tuned in downstream tasks and have achieved excellent results. The natural applications are sentence classification, sequence labeling, etc., which are more inclined to NLU tasks.

Auto-Regressive (AR) Models: The main task of AR models is to predict the next word based on what has been read in the text. This is the same as the classical language modeling approach. A representative of the AR models is the GPT family. Unlike the aforementioned AE language models, the infrastructures of AR are composed of the decoder part of Transformer, and a masking mechanism is used in the training phase so that the attention calculations can only see the content before a word. While it is possible to fine-tune such a PLM and achieve excellent results on many downstream tasks, its most natural application is NLG tasks.

Seq2seq Models: The sequence-to-sequence (Seq2Seq) models use both the encoder and decoder of Transformer for better model flexibility. Currently, the most representative models of this type include T5 [108] and mBART [80]. In principle, almost all pre-trained tasks used in AE and AR models can be adapted to the seq2seq models. Relevant research [108] has found that seq2seq models can achieve better performance. Moreover, a seq2seq model unifies the NLU and NLG tasks so that they can be solved under the same framework. It can be fine-tuned on a variety of NLG tasks such as translation and summarization, as well as NLU tasks that can be converted into a text2text form [108], including sentence classification, semantic similarity matching, etc.

Summary: Theoretically speaking, Auto-Encoder (AE) and Sequence-to-Sequence (Seq2Seq) models utilize bidirectional attention in Transformer, while Auto-Regressive (AR) language models rely on causal attention. Bi-attention models have been found to encounter low-rank problems [30], which may limit their expressive ability to some extent. On the other hand, causal attention possesses greater theoretical expressive power. Therefore, for complex tasks such as logical reasoning, or in a scene in which the language model is expected to serve as a backbone for Artificial Generative Intelligence (AGI), AR models with a significant number of parameters (e.g., GPT-3 [9] and GPT-4) are preferred. However, AR models also come with potential limitations for tasks such as fill-in-the-blank, content comparison, and text summarization, which often require the model to look back, analyze multiple pieces of content, or engage in extensive re-reading [9]. Therefore, AE and Seq2Seq models are often better choices for these tasks.

When it comes to controlled text generation using PLMs, most methods exploit the generative model, including AR and Seq2seq models, as the basis and guide them to generate the desired text. Generally, CTG tasks always treat the PLMs as a conditional generation model, and its formulation is consistent with the standard language model:

$$P(x_n|X_{1:n-1}) = p(x_n|x_1, x_2, \ldots, x_{n-1}). \tag{2}$$

Based on the pre-trained language model manifested above, the goal of conditional text generation can be formulated as

$$P(X|C) = \prod_{i=1}^{n} p(x_i|x_{<i}, C), \tag{3}$$

where C denotes the controlled conditions, which will be integrated into the PLM in a specific form, and X is the generated text that incorporates the knowledge encoded in the PLM and complies with the control conditions.

In the next section, we will review the main approaches to CTG using Transformer-based PLMs.

3 MAIN APPROACHES TO PLM-BASED CTG

From a generative point of view, PLMs have learned a variety of knowledge from a large-scale corpus that can help produce more fluent and a richer variety of text. This provides an effective way for natural language generation. However, the existing PLMs are essentially still black-box models like other deep neural networks, lacking interpretability and controllability of the text generation process. How to make good use of PLMs in text generation while realizing the controllability of the generative model has recently become a hot research topic. In this section, we provide a comprehensive review of the main approaches in this area from the perspective of the Transformer-based PLMs that are used for CTG.

3.1 Overview

The core idea of PLM-based CTG is to give the model a control signal in an explicit or implicit way to drive the generation of text satisfying the control conditions. According to how the control signal works, we have roughly divided the existing methods into three categories, each of which is further divided into several subclasses. An overview is given in Figure 4. The most direct way is to **fine-tune** the PLMs, which can perform the CTG task at a lower cost. The second way is to **retrain or refactor** the PLMs for CTG. In principle, this method could produce better results but may consume more computing resources and also face the problem of lacking labeled data. As the parameter size of PLMs increase rapidly, even fine-tuning has become resource-intensive. To tackle the problem, the third category of text generation methods, **post-processing**, that work on the decoder time, named **post-processing**, have emerged. In the post-processing methods, PLMs are

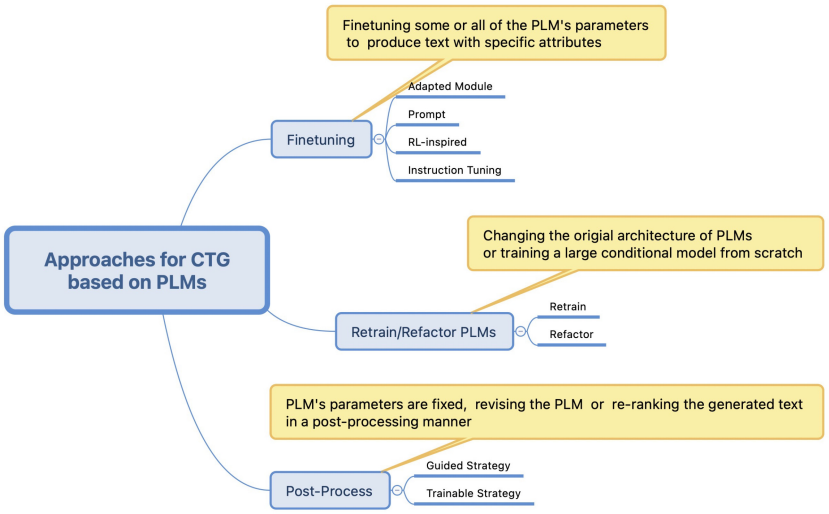


Fig. 4. An overview of the PLM-based CTG approaches. According to how the control signal works with the pre-trained language model, we have roughly divided the existing methods into three categories, each of which is further divided into eight subclasses.

always fixed, and the control signal works on the decoding stage. Such methods not only require less computation resources for training but also can guarantee a better quality of the generated text to some extent. As a consequence, increasing attention from the academic community has been paid to this direction in recent years. In the following sections, we will review the recent literature related to these three types of method in more detail.

3.2 Fine-tuning

This type of method aims to fine-tune part or all of the parameters of a PLM to produce text that satisfies the specific controlled conditions. As discussed in Section 3.1, "PLM + fine-tuning" has become a new paradigm in the general field of NLP. First, a large amount of training data (usually unlabelled samples) and model parameters are used to learn general knowledge from the data and encode the learned knowledge into a PLM. Then a domain/task-adapted model will be obtained to achieve the competitive performance by fine-tuning the PLM based on a small amount of labeled data for the specific downstream task.

This paradigm is also applicable to CTG, and a large number of related studies have been carried out. Recent work has found that fine-tuning PLMs on the target data, such as AMR-to-text [56, 91, 111] for dialogue generation, can establish a new level of performance. Because the conventional fine-tuning method is relatively concise and easy to understand, we focus on more advanced methods below.

Adapted Module: This method first constructs a task-related adapted network module around a PLM. Then, it is trained with the PLM on the target dataset just like usual fine-tuning. Auxiliary Tuning [156] introduces an extra condition modeling module based on the original PLM, which takes $X(x_n; C)$, the concatenation of control condition C and training text x_n, as input, and outputs logits in the vocabulary space. The auxiliary model is trained by adding its logits to the PLM's logits and maximizing the likelihood of the target task's output. In [112], an adapter module is added after the feed-forward sub-layer of each layer on both the encoder and decoder of the PLM, which can encode the graph structure into the PLMs without contaminating its original

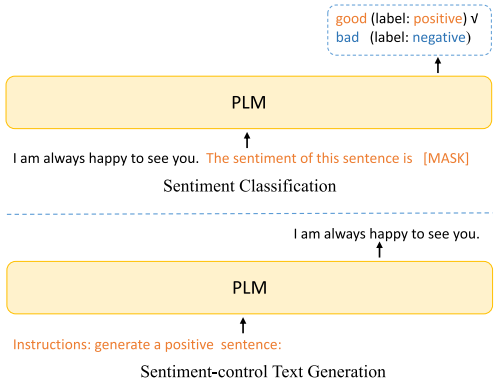


Fig. 5. An illustration of prompt learning. The top one shows an example of sentiment classification based on prompt learning, whereas the bottom one shows an example of sentiment-controlled text generation. The text in red is the templates, which could be either manually constructed by humans or automatically searched discrete/continuous tokens.

distributional knowledge. During the training stage, the PLM's parameters are frozen, and only the injected adapter is trainable. Avoiding the catastrophic forgetting problem while maintaining the topological structure of the graph, the model achieves the SOTA performance on two AMR-to-text benchmarks. On the controlled dialogue generation task, the idea of an adapted module is also applied. In [72], an adapter-bot for dialogue generation is proposed. The model builds a series of lightweight adapters on top of a PLM for dialogue generation, namely, DialoGPT [162]. The model allows for a high-level control and continuous integration of various control conditions for different conversational requirements (e.g., emotions, personas, text styles, etc.).

In summary, the adaptive modules essentially aim to bridge the gap between the controlled attributes and the PLMs while guiding the language model to generate text that meets the corresponding control conditions.

Prompt: A more effective way to use PLMs is to keep the training objective of the fine-tuning phase consistent with the original task from which the PLMs are derived [37]. This idea gives rise to the so-called *prompt-based approaches*. Take a sentiment classification task, for example. Suppose we need to recognize the sentiment of a sentence, e.g., *"I am always happy to see you"*. In contrast to the traditional approaches that encode the sentence into a set of vectors and then classify their sentiment through a fully connected layer, the prompt-based method will construct a set of templates, for example: (*"I am always happy to see you,* ***the sentiment of the sentence is [MASK]****"*), and then ask the model to predict the token *[mask]* according to the original training task for constructing the PLM. For sentiment-controlled text generation, the template could be regarded as the control prompts to instruct the PLM to generate desired texts. A specific illustration of prompt learning can be seen in Figure 5. The prompt-based approach has gone through various stages, from manual template construction [54] to automated search for discrete tokens [124] to continuous virtual token representations [63, 68]. Keeping most parameters of the PLM fixed in most cases and retaining generative capacity of the original PLMs, these methods have achieved great success in zero/few-shot scenarios.

From the CTG point of view, the prompt-based approach still applies. In [68], a method named "prefix tuning" is proposed, which freezes the PLM's parameters and back-propagates the error to optimize a small continuous task-specific vector called "prefix". The learned prefix, also called "prompt", can guide the PLM to generate the required text, thus enhancing the controllability

to a certain extent. The approach achieves impressive results on some generative tasks such as data-to-text. An extension of the model, P-tuning [63], serves a similar purpose. In contrast to prefix-tuning [68], P-tuning does not place a prompt with the "prefix" in the input; rather, it constructs a suitable template to prompt the PLM. The template is composed of continuous virtual tokens obtained through gradient descent. Based on prefix-tuning, Qian et al. [103] leverage contrastive learning to train attribute-specific vectors. In contrast to the vanilla prefix tuning, in which each prefix is trained independently under the attribute-specific corpus, they take into consideration the relationship among attribute prefixes and train multiple prefixes simultaneously, thereby boosting the control performance. DisCup [158] provides a new prompt-based alternative for attribute-controllable generation, which uses the attribute-discriminator to assist prompt tuning. This allows the learned control prompt to absorb the information of inter-attribute knowledge, achieving the state-of-the-art attribute control performance. In view of the promising effect and the parameter-efficient structure features of the prompt tuning in CTG, prompt-based multi-attribute control approaches have been proposed. For example, Tailor [152] explores two strategies, including a non-training method based on the concatenation of single-attribute prompts and a training method using an attribute connector, for prompt-based multi-attribute CTG.

In order to tackle the problem that the distance between the prompt and the next predicted token correlates negatively with the prompt's influence power, Zou et al. [172] propose a method called Inverse Prompt. The main idea is to use generated text candidates from the PLM to inversely predict the prompt (topic, poetry name, etc.) during beam search, so as to enhance the relevance between the prompt and the generated text and achieve a better controllability. However, the generation process requires the reverse prediction for each candidate token, leading to an increased computation cost. According to our actual tests based on the provided source code, it takes up to around 10 minutes to generate a seven-word rhyming poem, making it difficult to be applied in real application scenarios.

More recently, in order to address the challenge of fine-grained CTG, an encoder–decoder architecture based on a pair of GPT-2 models is introduced [10]: non-residual prompting. It enables intermediate text prompts at arbitrary timesteps of the generative PLM. Specifically, the proposed method uses an auxiliary prompt encoder, composed of a trainable GPT-2, to guide another generative language model towards certain constraints, including themes, sentiment, keywords, etc. The generative language model (i.e., GPT-2) is always fixed during generation, and the different prompt instructions can be used at different timesteps, enabling fine-grained CTG. Non-residual prompt trends to be versatile and shows promise towards the unified controllable text generation. However, we hold a critical opinion on it, as it lacks a systematic comparison with the natural encoder–encoder architectures such as T5 [108] and Bart [80], and its training process is complex so that it is not parameter'efficient.

To summarize, most of the prompt-based methods show a certain degree of versatility. From the CTG perspective, this kind of method essentially uses the characteristics of PLM in its pre-training stage to guide the PLM to generate constrained text by selecting an appropriate prompt in the fine-tuning stage to achieve the purpose of controllability.

Reinforcement Learning (RL)–inspired Approaches: The core motivation of this type of method is to feed back whether or how the control conditions are achieved as a reward to the fine-tuning of the PLM. Ziegler et al. [171] use reinforcement learning to fine-tune the PLMs, with a reward model trained from human preferences. First, it initializes a policy $\pi = \rho$, where ρ denotes a PLM such as GPT2. Given a dataset $\mathcal{D} \in (X, Y)$, the goal is to fine-tune π so that it can approximate the distribution of the data $\mathcal{D}$. This is done using RL by optimizing the expectation of the reward:

$$\mathbb{E}_{\pi}[r] = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(\cdot|x)}[r(\pi(x), y)]. \tag{4}$$

Then, the reward model r is trained based on the sample (x, y_0, y_1, y_2, y_3) via $x \in \mathcal{D}$, and y_i is generated from $p(y_i|x)$. Human labelers are required to choose the human-preferred sentence from (y_0, y_1, y_2, y_3). To prevent π from moving too far away from the original PLM p for ensuring the fluency of the generated text to the greatest extent, a penalty item is added to the reward function during the actual process of fine-tuning π:

$$R(x, y) = r(x, y) - \beta KL(\pi, p), \tag{5}$$

where $R(x, y)$ is the re-defined reward function, β is the regular coefficient, and $KL(\pi, p)$ aims to ensure that the two distributions are as close as possible.

Liu et al. [79] propose a data augmentation approach, which uses reinforcement learning to guide the GPT-2 model to generate texts towards a specified conditional direction (i.e., the target class). Specifically, an additional RL stage is added between the softmax and argmax functions of GPT2. Then, the parameter of the PLM's hidden states θ is updated towards the target label according to the signal of the RL reward. The generated texts are regarded as augmentation data to help improve the classification performance. Moreover, Stiennon et al. [130] use an RL-based approach on the task of English summarization, which fine-tunes the PLM by combining with human feedbacks.

Reinforcement learning is also applied for controllable story generation. Tambwekar et al. [132] design a reward-shaping technique that produces intermediate rewards at all different timesteps, which are then back-propagated into a language model in order to guide the generation of plot points towards a given goal. It should be noted that the above work is carried out on a language model based on long short-term memory (LSTM), but its principles are applicable to the subject of this article. Thus, we have included it here.

In summary, the idea of applying reinforcement learning to PLM-based CTG is natural. The central challenge is to ensure that the PLM is optimized towards the RL's rewards while maintaining the fluency of the generated text. To address this challenge, the key is to achieve a better balance between these two aspects.

Instruction Tuning: Recently, a new PLM-based CTG paradigm, instruction tuning, has become popular. Instruction tuning provides an avenue to align the language models with user intents, i.e., controlling language models to generate the content that complies with human instructions.

Google Research proposes FLAN [144], which stands for **F**ine-tuned **La**nguage **N**et. It involves fine-tuning a large language model on a mixture of more than 60 NLP datasets, where each task is expressed through natural language instructions. The results demonstrate that language models are capable of performing tasks described purely through human instructions and can generalize to previously unseen tasks through instruction tuning. Continuing the work of FLAN, Chung et al. [20] further explore scaling up the number of tasks and model size beyond what FLAN has achieved. They fine-tune the language models on the dataset mixed with chain-of-thought data, showcasing the strong few-shot performance of instruction tuning.

InstructGPT [94], a most notable recent work, utilizes instruction tuning to control the language model and generate desired human-like content. It starts with collecting a dataset of labeled demonstrations of the desired model behavior, which are then used as instructions to fine-tune GPT-3. This allows for controlling the model to generate answers that align with human expectations. In terms of the optimization algorithm, InstructGPT leverages reinforcement learning from human feedback [130, 171], as discussed in the previous section. The results demonstrate that fine-tuning with human feedback is a promising approach to aligning language models with human intents, leading to an improved performance in truthfulness and a reduction of toxic output.

In summary, instruction tuning enables PLMs to understand human intents in a natural language format, offering a promising approach for more general and effective CTG. However, instruction tuning requires careful design of human-labeled prompts. How to fully and safely align the human instructions with PLMs remains an open problem [94], which demands further exploration.

3.3 Retraining/Refactoring

According to the characteristics of a specific downstream task, it is also feasible to change the original architecture of PLMs or retrain a large conditional language model from scratch. This kind of approach is promising to substantially improve the quality and controllability of text generation, but is limited by increased computing resource consumption and the lack of sufficient labeled data.

CTRL [58] is an early attempt in this direction. It trains a language model conditioned on a variety of control codes. The network model used in this approach is also the commonly used Transformer, and a piece of control code (domain, style, topics, dates, entities, relationships between entities, etc.) is added in front of the text corpus. That is, it transforms the original language model $p(x_i \mid x_{<i})$ into $p(x_i \mid x_{<i}, c)$. A language model with 1.63 billion parameters is retrained on a 140 Gb corpus. Another contribution of this work is to propose a new top-k sampling algorithm:

$$p_i = \frac{\exp\left(x_i/(T \cdot I(i \in g)\right)}{\sum_j \exp\left(x_j/(T \cdot I(j \in g)\right)} \quad I(c) = \theta \text{ if c is True else 1}, \tag{6}$$

where g is a list of generated tokens, p_i is the probability distribution for the next token, and the introduction of $I(c)$ reduces the probability of words that have already appeared.

Zhang et al. [163] propose POINTER, an insertion-based method for hard-constrained (i.e., making specific words appear in generated text) text generation. In contrast to the auto-regressive method, such as GPT-2, this method modifies the structure of the Transformer so that it can generate text in a progressive manner. Specifically, given certain lexical constraints, POINTER first generates the constrained words to satisfy the control conditions, then more detailed words are inserted at a finer granularity between those words. The above process iterates until the entire sentence is completed. This kind of method can ensure that the generated sentences meet the lexical constraints. However, the model needs to be trained from scratch on the large-scale corpus, and the fluency of the generated sentences is not as good as the auto-regressive model in most cases.

Similar to the afore-mentioned insertion-based method [163], a lexically constrained text generation framework called Constrained BART (CBART) is proposed [45]. This approach also adopts the progressive insertion/replacement for text generation, yet without modifying the Transformer's architecture. Concretely, based on the pre-trained model BART, it divides the generation process into two steps. First, a token-level classifier is added on BART's encoder to predict where to replace and insert. Then, the predicted results are regarded as signals to guide the decoder to refine multiple tokens of the input in one step by inserting or replacing tokens before specific positions. Different from the normal way of generating texts step by step, the decoder predicts all tokens in parallel to accelerate the inference. Although CBART does not need to reconstruct the architecture of PLMs, the training and inference processes are different from the original pre-training tasks, which can lead to a negative impact on the quality of text generation.

CoCon (Content-Conditioner) [14] introduces a conditional control module in addition to the original PLM, which can realize the precise control of the generated text at the word and phrase levels. In terms of model architecture, the approach injects a control block into the GPT model and provides the control code as a separate input. In order to tackle the problem of lacking labeled data, it adopts self-supervised learning and constructs four different loss functions, including Self-Reconstruction Loss, Null Content Loss, Cycle Reconstruction Loss, and Adversarial Loss. The core of these self-supervised losses is to use one part of a piece of text as a control condition,

leaving the rest for the model to refactor, so that the model can learn to generate specific text conditioned on the control code. The experimental results show that CoCon can incorporate the condition content into the generated texts and control the high-level text attributes in a more flexible way. Similar to CoCon's idea of injecting an additional controlled module to an existing PLM, Wang et al. [143] propose a Mention Flags (MF) module, which is injected into the decoder of Transformer, to achieve a higher level of constraint satisfaction. The MF is designed to trace whether a lexical constraint has been realized in the decoder's output. It is formally represented as mentioned status embedding injected into the decoder of Transformer to provide a signal to encourage the generative model to satisfy all constraints before generation.

This type of approach is also used for the task of controlled dialogue generation. Zheng et al. [167] propose a PLM-based method to build a personalized dialogue agent. The whole framework is in an encoder–encoder (Transformer) fashion, and its initial parameters inherit from an existing PLM model. The personalized information is represented as attribute embedding, which is added into the encoder to capture rich persona-related features when modeling dialogue histories. Further, an attention routing network is added to the decoder to incorporate the target persona in the decoding process while maintaining the trade-off of the historical dialogue information dynamically. To solve the problem of lacking labeled data in conditional dialogue generation, a multi-task learning framework is proposed [157] that utilizes both conditional labeled dialogue data and non-dialogue text data. Based on a condition-aware Transformer block (reconstructed from the original Transformer), three sub-tasks are designed based on the existing PLM, namely, conditional text generation based on labeled dialogue data, conditional conversation encoder, and conditional dialogue generation task based on non-dialogue text to optimize the model simultaneously. Persona and topic-controlled experiments are conducted under the scenario of dialogue generation, and the results show that this approach achieved a then-state-of-the-art performance.

In summary, the refactoring or retraining approaches are more convenient to use, but may lose the original PLM's versatility to some extent. As for the methods that need retraining, they may face the dual challenges of increased computation cost and the lack of large-scale labeled data.

3.4 Post-Processing

When the number of parameters of a PLM increases, the model has memorized more and more knowledge and patterns, allowing it to achieve competitive results even without fine-tuning in many NLP tasks [9]. In the realm of controlled text generation, the idea of fixing the PLM's parameters first and re-ranking the generated text in a post-processing manner becomes achievable and promising. Figure 6 illustrates the use of a post-process module for sentiment-control text generation.

The most realistic idea about the post-processing method is to use some standard decoding algorithms in text generation, e.g., the Greedy search, constraint beam search [2], Top-k sampling [32], Nucleus sample [46], etc. The approaches discussed below can be seen as an extension of them for CTG tasks. They are grouped into two categories: guided strategies and trainable strategies.

Guided Strategies: This type of method decouples the PLMs for text generation and the post-processing module, and the post-processing module guides the PLM to generate conditioned text only in the inference stage.

A representative method of this type is PPLM [24]. It first trains an attribute discriminant model and then uses it to guide the PLM to generate the corresponding text. In this work, the attribute model is a simple classifier, consisting of a user-specified bag of words or a single learning layer whose parameters are 100,000 times less than the PLM. During the text sampling process, it requires a forward and backward process in which the gradient from the attribute model drives the hidden activation of the PLM to guide the target text generation. PPLM does not need to change

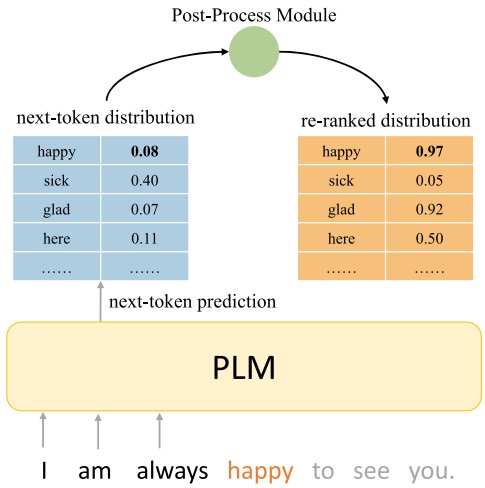


Fig. 6. The scheme of the post-process approaches for CTG. During the generation, the parameters of the PLM are fixed. Given the input prompt "I am always", the PLM will produce the original next-token distribution; the post-process module aims to re-rank the token distribution, allowing the PLM to choose the sentiment-desired tokens (i.e., significantly increasing the probability of "happy"), so as to achieve the positive sentiment control.

the structure or retrain the PLM, and it is able to achieve a significant improvement in attribute alignment. However, it causes a slight decrease in text fluency measured with the metric of PPL (Perplexity [62, 74]).

MEGATRON-CNTR [150] is a controllable story generation framework that combines external knowledge and PLM. Given a story context, a predictor is used to get a set of keywords for the next sentence. Then, a knowledge retriever is introduced to get external knowledge–enhanced sentences from an external knowledge base according to the keywords. Next, a ranker is trained to choose the most relevant knowledge-enhanced sentences, which are later fed into the generator of PLM (GPT-2) with the story context to get the next sentence. The entire process is repeated until completing a story. Human evaluation results show that up to 91.5% of the generated stories are successfully controlled by the keywords. In this framework, GPT-2 is independent of the other modules and generates context-relevant sentences using the introductory text provided by MEGATRON-CNTR's component as input, without any adaptation in training.

Hua and Wang [49] propose a content-controlled text generation framework called FAIR. It uses the BERT [27] model to automatically construct a content plan, including keyword assignments and their corresponding sentence-level positions. After that, the BART [80] model is applied, without structure modification, to fill the masked tokens appearing in the generated text template. Finally, an iterative refinement algorithm that works within the sequence-to-sequence (seq2seq) models is designed to improve generation quality with flexible editing. The reported experimental results show that FAIR can significantly improve the relevance and coherence between the key phrases and the generated texts.

Additionally, a series of discriminator-guided approaches have been developed that train an attribute discriminator to help the PLM select text for the specific attributes at the decoding stage. Adversarial Search [117], inspired by GAN (Generative Adversarial Network), trains the discriminator to distinguish human-created text from the machine-generated text. The discriminator predicts a label for each token instead of for the entire sequence. Its logit probability is added to the

score to guide sampling towards the human-written style. For the tokenizer with 10,000 words, decoding using a discriminator to classify each token is time-consuming. Aimed at solving this problem, GeDi [62] trains a small class-conditional language model (CC-LM) as generative discriminators to guide the generation from large PLMs (GPT-2 and GPT-3). Specifically, a CC-LM is trained and formulated as the following equation:

$$P_\theta\left(x_{1:T} \mid c\right)=\prod_{t=1}^{T} P_\theta\left(x_t \mid x_{<t}, c\right), \tag{7}$$

where c is the control code and T is the length of generated text. GeDi assumes that there is a CC-LM with the desired control code c and an undesired or anti-control code $\bar{c}$. It then uses the contrast between $P_\theta\left(x_{1:T} \mid c\right)$ and $P_\theta\left(x_{1:T} \mid \bar{c}\right)$ to guide sampling from the original PLM. A contrast mechanism is designed to compute the probability that every candidate token x_t belongs to the desired class, given by $P_\theta(C \mid x_t, x<t)$:

$$P_\theta(c \mid x_{1:t})=\frac{P(c) \prod_{j=1}^{t} P_\theta(x_j \mid x_{<j}, c)}{\sum_{c' \in\{c, \bar{c}\}} \prod_{j=1}^{t} P(c') P_\theta(x_j \mid x_{<j}, c')}, \tag{8}$$

where $P(c)$ and $P(c')$ are biased parameters that could be learnt or set manually as a hyperparameter. During the generation step for a token x_t, Equation (8) is multiplied with the conditional probability $P_{LM}\left(x_t \mid x_{<t}\right)$ of the original PLM via the Bayes rule:

$$P_w\left(x_t \mid x_{<t}, c\right) \propto P_{LM}\left(x_t \mid x_{<t}\right) P_\theta\left(c \mid x_t, x_{<t}\right), \tag{9}$$

where $P_w\left(x_t \mid x_{<t}, c\right)$ is regarded as the final probability for text generation. Since the calculation of Equation (8) only needs two parallel forward passes of CC-LM, the generation efficiency is greatly improved.

Inspired by the GEDI [62], a series of similar approaches have emerged. DEXPERTS [74] reranks the predictions of the PLM based on expert (and anti-expert) opinions during the decoding stage to steer the language model towards generation of the desired text. FUDGE [151] learns an attribute predictor that operates on a partial sequence to adjust the original PLM's probabilities, achieving an improved performance on the tasks of couplet completion in poetry, topic control in language generation, and formality change in machine translation. Plug-and-Blend [73] extends the GEDI model to controlled story generation by introducing a planner module.

More recently, Pascual et al. [96] proposed a simple yet efficient plug-and-play decoding method called K2T, which does not even need a discriminator. Specifically, given a topic or keyword that is considered a hard constraint, K2T adds a shift to the probability distribution over the vocabulary towards the words that are semantically similar to the target constraint word. The shift is calculated based on word embedding. Although the K2T is intuitive, the shift added to the probability distribution of vocabulary may be too rough and cause the generated texts to fall short in fluency.

Mireshghallah et al. [89] propose a score-based controllable text generation framework called Mix and Match, which regards the task of generating attribute-specific text as a Metropolis-Hastings sampling process. Specifically, the proposal samples are first produced by the MLM model (i.e., BERT). Then, an energy-based model, which is a linear combination of scores conditioning context from the different black-box experts that represent fluency, the control attribute, and faithfulness, respectively, is used to accept/reject the proposal sample. Those two steps are iterated until the desired texts are obtained. The framework is training free, without any fine-tuning or structural assumptions. Nevertheless, Mix and Match is an iteration-based method. It takes almost 11 seconds to generate a sequence of length 20, which reduces the practical value. Similarly, COLD Decoding [106], an energy-based and iterative CTG approach, formulates the controlled text generation task as sampling from an energy-based model using Langevin Dynamics, without the need

for any task-specific fine-tuning. Like the Mix and Match method mentioned earlier, COLD Decoding also suffers from the efficiency issue in text generation.

To sum up, the idea of "guided strategies" is simple and flexible. The main advantage of this approach lies in the separation of the post-processing module from the model. When the number of parameters of the PLM increases, such advantage becomes more apparent. However, post-process requires multiple iterations to achieve better control performance, resulting in excessive time costs.

Trainable Strategies: The Trainable Strategies also work in the inference phase, but in contrast to the Guided Strategies, the extra processing module needs to be trained jointly with PLM, whose parameters are fixed. Compared with the prompt-based approaches, the module controls PLM's generation process noninvasively, without disturbing the model's original textual stream.

An Energy-Based Model (EBM) [25] is proposed to guide the PLM to generate desired text. The generative model is formalized as follows:

$$P_\theta(x) \propto P_{LM}(x) \exp\left(-E_\theta(x)\right), \tag{10}$$

where $P_{LM}(x)$ is a local normalized language model whose parameters are frozen during training and $E_\theta(x)$ is an energy function that is aimed at steering the joint model $P_\theta(x)$ towards the desired data distribution. The Noise Contrastive Estimation (NCE) algorithm is used to train the model to cope with the intractability issue of the energy model. Experiments show that the proposed method yields a lower perplexity compared with locally normalized baselines on the task of generating human-like text. The use of large-scale PLM makes this method possible because the quality of the generated text from the joint model relies heavily on the quality of the underlying language model.

Furthermore, since RL-based methods may lead to the problem of "degeneration" in the sense of producing poor examples that improve the average reward but forgo the coherence and fluency, a distributional approach for controlled text generation [59] was proposed to solve the problem. It uses the Energy-based Model (EBM) to represent the point-wise and distributional constraints in one go:

$$p(x) \doteq \frac{P(x)}{Z}, \tag{11}$$

$$Z \doteq \sum_x P(x), \tag{12}$$

$$P(x) = a(x)e^{\sum_i \lambda_i \phi_i(x)}, \tag{13}$$

where $p(x)$ is the desired normalized distribution, Z is the normalized term, $\phi_i(x)$ represents the constraint judgment function (point-wise it means 0 or 1, distributional-wise it may be a continuous value between 0 and 1), and λ_i is the corresponding coefficient estimated by using Self Normalized Importance Sampling (SNIS), $a(x)$ is the original PLM. As it is hard to calculate $p(x)$ directly, a method similar to variational inference is adopted to approximate the distribution, by first initializing a policy $\pi = a(x)$ and then minimizing the cross entropy between the policy π and the desired normalized distribution $p(x)$:

$$CE\left(p, \pi_\theta\right) = -\sum_x p(x) \log \pi(x). \tag{14}$$

The optimization process adopts the KL-Adaptive distributional policy gradient (DPG) algorithm to make π approximate the model $p(x)$ that satisfies the constraints. The method unifies the point-wise and distributional-wise constraints in a single framework, and the experimental results show its superiority in satisfying the constraints while avoiding degeneration. However, this approach suffers from the high computational cost, which needs to be addressed in the future.

Table 3. A Summary of the Surveyed CTG Methods Divided into Three Main Categories and Eight Subcategories

Method	Main characteristics	Subcategory	Typical References
Fine-tuning	standard training; efficient inference; higher text quality; weaker controllability	Adapted Module	[72, 112, 156]
		Prompt	[54, 63, 103, 124] [10, 68, 152, 158]
		Reinforce Learning	[79, 94, 130, 132, 171]
		Instruction Tuning	[94]
Refact/Retrain	computationally expensive training; higher text quality; better controllability	Retrain	[58],[163]
		Refact	[14, 45, 143, 157, 163, 167]
Post-process	efficient training and inefficient inference; lower text quality; better controllability	Guided Strategy	[24, 62, 74, 117, 150, 151] [73, 89, 96, 106]
		Trainable Strategy	[25, 59]

We list the main characteristics for each categor from a macro perspective, and representative references for each subcategory.

Generally speaking, Trainable Strategies require the post-process module to be jointly trained on the basis of PLM, and realize controllable text generation by adjusting the original probability distribution of PLM to the desired data distribution using the trained post-process module. This type of method builds upon the probability modeling theory and thus has a good theoretical basis, yet it is still at the early stage and needs to resolve the issues related to computational efficiency and text quality.

3.5 Summary

In this section, we divide the current PLM-based CTG approaches into three categories according to how PLMs are used. For each category, we analyze its main principles, process, and methods. A summary of the surveyed CTG models is shown in Table 3. "Fine-tuning" is a more general type of method that has been widely used in both NLU and NLG tasks. How to make full use of the power of PLMs in specific tasks is still a hot research topic for future research. The retraining or refactoring approaches typically involve high training costs and the lack of large-scale labeled data. To overcome these limitations, combining the general pre-trained models with semi-supervised or self-supervised learning to build a pre-trained model dedicated to CTG would be a feasible direction for future research.

The emergence of post-processing methods is rooted in the powerful text-generation capabilities of PLMs. This kind of method generally assumes that the PLMs can produce high-quality text and then uses a post-processing module as a filter to screen the desired type of text. Since the post-process modular is usually decoupled from the PLMs, most current decoding-time approaches (post-process) are still computationally expensive (i.e., with longer inference time or additional parameters), and the quality of generated text can be low. However, it has some promising advantages, because the parameters of the PLMs do not need to be retrained, thus greatly saving computing resources for the model training stage. In recent years, the scale of pre-trained models has been getting larger, and their mastery of language knowledge is getting more comprehensive. At the same time, the sheer size of parameters makes the PLMs resource-intensive to fine-tune and retrain. The above-mentioned trends coincide perfectly with the advantages of the "post-process" methods. Thus, it has a great potential for future research and development.

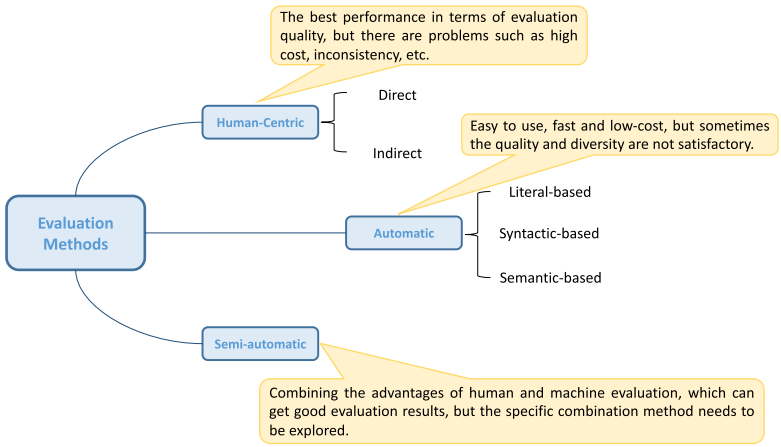


Fig. 7. A categorization of general NLG evaluation methods.

4 EVALUATION METHODS

The performance of an NLG model is reflected by suitable evaluation metrics. CTG is slightly different from the general NLG tasks due to the need to fulfill the controlled elements. Therefore, CTG is concerned about not only the quality of the generated text but also the satisfaction with the controlled elements. As a consequence, we usually use both general and CTG-specific metrics to evaluate a CTG model.

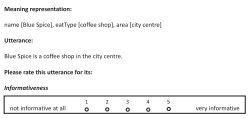
4.1 General NLG Evaluation Metrics

For any CTG model, it is essential to evaluate the quality of generated text in general aspects such as (1) **fluency** —how fluent the language in the output text is [12, 31], (2) **factuality** —to what extent the generated text reflects the facts described in the context [46, 146], (3) **grammar**: —how grammatically correct the generated text is, and (4) **diversity** —whether the generated text is of a diverse range of types or styles. The ways of measuring these general evaluation aspects can be divided into three categories based on who performs the assessment: human beings or the machine (as shown in Figure 7).

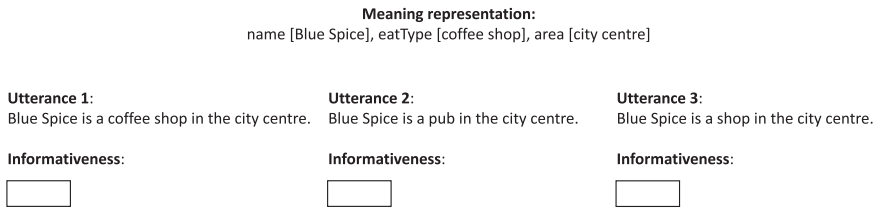
4.1.1 Human-centric Evaluation Metrics. Human beings create Natural Language as a crucial form of human communication. Thus, humans are the best evaluators of the natural language texts generated by NLG systems. We call the evaluation metrics that involve human assessors only *human-centric evaluation metrics*, which can be roughly divided into two types:

Direct evaluation. In this type, human assessors judge the quality of the generated texts directly. A simple way is to make a binary decision, i.e., good or bad, and a more complex way is to use finer-grained decisions: e.g., Likert scale as shown in Figure 8(a), and RankME in Figure 8(b), etc. [12, 46, 93].

Indirect evaluation. In contrast tp the direct evaluation, indirect evaluation is done by measuring the effect of the generated text on downstream tasks, from either a user's perspective (such as whether or not it leads to an improved decision-making or text comprehension accuracy [39], typically through a User Task Success evaluation), or from a system's perspective [3, 26], such as the performance of a dialogue system through the System Purpose Success evaluation.



(a) An example of Likert scale [13]. In this example, human annotators need to rate the informativeness (an index measuring whether the generated text provides all the useful information from a given meaning representation) with a score ranging from 1 to 5.



(b) An example of RankME [13]. Human annotators need to give a score of the informativeness in a given range for each utterance in this example based on the given meaning representation.

Fig. 8. Examples of human-centric evaluation metrics include the Likert scale and RankME.

4.1.2 Automatic Evaluation Metrics. Automatic evaluation metrics for NLG usually compare the similarity of the NLG model **generated texts** G to the corresponding **reference texts** R (i.e., human-written texts) in the benchmarking datasets. We divide the similarity measures into three categories: lexical-based, syntactic-based, and semantic-based.

Lexical-based Metrics. The lexical-based metrics measure the similarities between basic lexical units (e.g., words or phrases) across the pair of sentences, which are then aggregated into an overall sentence-level similarity.

BLEU [95] is a commonly used metric in natural language processing tasks to evaluate the similarity between the generated text from an NLG model and the corresponding reference text. Specifically, BLEU counts the n-gram matches in the generated text against the reference text. For convenience, we use BLEU-n to represent BLEU with respect to n-grams, as follows.

$$\text{BLEU-n} = \frac{\sum_{t \in G} \sum_{n-gram \in t} Count_{match}(n-gram)}{\sum_{t \in G} \sum_{n-gram \in t} Count(n-gram)}, \tag{15}$$

where t is a piece of generated text to be compared and *match* means that an n-gram in t also appears in the reference text. The larger BLEU-n is, the better the quality of the generated text.

Self-BLEU [170] is proposed as a metric to measure the diversity of the generated text. Since BLEU can measure the similarity of two different texts, self-BLEU calculates the BLEU score between each pair of generated texts and takes the averaged BLEU score to represent the diversity of all generated texts. The generated texts with a higher diversity have a lower self-BLEU score.

ROUGE [71] is the abbreviation of *Recall-Oriented Understanding for Gisting Evaluation*. It compares the generated text with a group of reference texts, counts the number of overlapping basic

units (n-grams), and obtains the corresponding score to measure the similarity between the automatically generated texts and the reference texts. The formula of *ROUGE* is as follows:

$$\text{ROUGE-n} = \frac{\sum_{t \in G} \sum_{n-gram \in t} Count_{match}(n-gram)}{\sum_{t \in R} \sum_{n-gram \in t} Count(n-gram)}, \tag{16}$$

Comparing Equations (15) and (16), the only difference between BLEU-n and ROUGE-n lies in the denominator. BLEU-n is focused on precision and the denominator is the total number of n-grams in the generated texts. ROUGE-n is recall-oriented, so the denominator is the total number of n-grams in the reference texts. A larger ROUGE-n indicates a better recall-oriented quality.

Perplexity (PPL) is a metric to measure how well a probabilistic language model predicts a sample. Essentially, a probabilistic language model is a probability distribution over a given text, i.e., probability of the $(n+1)-th$ word given the first n words of the text. For the task of NLG, a language model is trained on the reference texts, and is used to predict the generated text. The higher prediction probability indicates the better quality of the generated text. In practice, PPL takes the reciprocal form as follows:

$$PPL = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{p(w_i \mid w_1 w_2 \ldots w_{i-1})}}, \tag{17}$$

where n is the number of words in the generated text, w_i is the $i-th$ word in it, and $p(w_i)$ is the probability of w_i in the language model trained with reference texts. Generally speaking, The smaller the PPL value, the better fluency of the generated text. However, the lower PPL also implies more repetitions. Hence, another criterion is that the closer the PPL value of the generated text is to the human-written text, the better the fluency [46].

In addition, a variant of PPL, called **Reverse PPL** [123], has also been used to measure the diversity of the generated text by training a language probability model on the generated texts and calculating the PPL score on the reference texts. A smaller Reverse PPL means that the generated texts are quite different from the reference texts, indicating a higher diversity.

Distinct-n [64] is an n-gram–based metric and applies to some scenes (such as dialogue and advertisement generation), where the diversity of generated texts is pursued. The greater the Distinct-n value, the higher the diversity. It is formulated as follows:

$$\text{Distinct-n} = \frac{Count(unique\ n-gram)}{Count(n-gram)}, \tag{18}$$

where the numerator is the number of unique n-grams that appear in the generated texts and the denominator is the total number of n-grams in the generated texts.

Syntactic-based Metrics. Syntax is related to the grammatical arrangement of words in a sentence. A common syntactic-based metric is **TESLA** [75], which is the abbreviation for *Translation Evaluation of Sentences with Linear-programming-based Analysis*. In general, there are three variants of TESLA [22]: TESLA-M (minimal), TESLA-B (basic), and TESLA-F (full). TESLA-M uses N-gram matching and basic linguistic analysis such as lemmatization, part-of-speech tagging, and WordNet synonym relations. TESLA-B, a new configuration, adds bilingual phrase tables to model phrase synonyms. The most advanced version, TESLA-F (also called TESLA), goes further by incorporating language models and a ranking support vector machine instead of simple averaging.

Semantic-based Metrics. Semantic-based metrics aim to handle the evaluation of texts that are lexically different but have a similar semantic meaning. Compared with the lexical-based and syntactic similarity, the semantic similarity requires consideration of more information and is more difficult to measure. In recent years, PLM-based semantic evaluation methods have emerged,

which aim to evaluate the generated text using the trained PLMs inversely. A typical example is BERTscores [161], which replaces the n-gram overlaps defined in BLEU with the embeddings from BERT to learn a semantic-awareness metric. Similarly, Bertr and YiSi [87] take advantage of BERT embeddings to capture the semantic similarity between the generated text and its reference. There are also approaches that try to fine-tune PLMs for text quality estimation [81, 110, 169]. Particularly, Sellam et al. [118] propose a task-specific pre-trained model, BLEURT, for text assessment. BLEURT first adds random perturbations to Wikipedia sentences to construct millions of synthetic examples. Then, a BERT-based pre-trained model is trained on several lexical- and semantic-level supervision signals with a multitask loss. The experiments show that BLEURT benefits from pre-training and is robust to both domain and quality drifts. MAUVE [99] is an automatic metric to compare human-written and machine-generated texts. Specifically, MAUVE compares the machine-generated text distribution to that of human-written text, which are embedded by a PLM, using divergence frontiers, to reveal two different errors of the NLG model: (1) the model assigns high probability to sequences that do not resemble human-written text, and (2) the generative model fails to yield diverse samples as human-written texts.

4.1.3 Semi-automatic Evaluation Metrics. When faced with more diverse tasks, such as story generation and open-domain dialogue generation, the automatic evaluation methods turn out to be less ideal, because they can mainly evaluate the surface-level similarity between the reference and target sentences. Recognizing that humans can better distinguish a more diverse range of features, such as fluency and grammar, semi-automatic evaluation that combines automatic and human-centric evaluation methods has been developed, to get more reliable evaluation results.

Lowe et al. [84] encode the context, the generated text, and a reference text into vectors ($\mathbf{c}$, $\mathbf{g}$, and $\mathbf{r}$, respectively) using a hierarchical RNN encoder. Then the dot-product operation is adopted to transfer the vectors into a score ($score(\mathbf{c}, \mathbf{r}, \mathbf{g}) = (\mathbf{c}^T M \mathbf{g} + \mathbf{r}^T N \mathbf{g} - \alpha)/\beta$). Finally, this score is made as close as possible to the human judged score. A model that uses human judgments as labels can ensure good consistency with human judgments, but sometimes it may be too conservative and lacks diversity due to the quality of human evaluators.

Without using a human-judged score as the label, we can simply calculate the perplexity of a probabilistic model first and then let humans evaluate the beam-searched outputs. Specifically, Hashimoto et al. [43] encode the human judgments and model outputs into the same space by using the same encoder. Then, a discriminator is trained to distinguish whether the text is generated by a model or by a human evaluator. Finally, the leave-one-out error of the discriminator is computed. By doing so, we can preserve the diversity and quality of the generated text at the same time.

4.2 CTG-specific Evaluation

In addition to the above three categories of general NLG evaluation metrics, the CTG task demands additional evaluation metrics that take into account the controlled elements, i.e., whether a CTG model has fulfilled the specific controlled conditions. According to the definition of the controlled elements in Section 2.1, we can divide CTG-specific evaluation methods into two kinds of automatic evaluation methods. Human evaluation metrics for CTG are also discussed.

Semantic consistency metrics. This kind of evaluation method generally corresponds to the semantic control conditions. We first need to construct a training set with positive (samples satisfying the control conditions) and negative (samples not satisfying the control conditions) samples. Then, a classifier is trained to identify whether the model generates the controlled text that is semantically consistent with the controlled attributes, e.g., sentiment, topic [24, 103], style [17, 38], and toxic [62, 74, 158]. **Accuracy** is always used to measure the semantic consistency performance of the CTG model. It is calculated as the number of test samples correctly according with the

controlled elements divided by the number of test samples. Recently, CTRLEVAL [57], a **PLM-based** and reference-free method, was proposed for a training-free evaluation of CTG models. Specifically, CTRLEVAL formulates three aspects — coherence, consistency, and attribute relevance — to evaluate the generated text. Each aspect is defined as multiple text-infilling tasks with the base PLM, from which the ensemble of generation probabilities forms the final evaluation results. This approach does not need any labeled data and achieves higher correlations with human judgments.

Rule-based metrics. When it comes to the structurally and lexically controlled text generation tasks, we can use certain rules to judge how well the generated text conforms to the pre-defined controlled elements and then count the number of test samples that satisfy the control conditions as the final evaluation results. For example, **Coverage** can be used to measure the effectiveness on the lexical-constrained text generation tasks [10, 45, 143, 163], e.g., by calculating the average percentage of input keywords that are present in the generated text. **Success Rate** is used to measure the degree of matching between the generated text and the given structural control elements (i.e., Parts-of-Speech [POS], Syntax Spans, and Syntax Tree). For this purpose, an external tool for sentence structure extraction, such as oracle POS tagger and off-the-shelf syntax parser, is applied to the generated texts [69]. Moreover, some general text evaluation metrics can also be used for CTG-specific evaluation. For instance, the lexical-level overlap between the reference text containing the control elements and the generated text can be used to reflect how well the control conditions are met in the task of table/graph to text [102, 111, 112, 128, 131, 165] and story generation [33, 40]. Thereby, general metrics such as BLEU-n and ROUGE-n could also be regarded as CTG-specific evaluation metrics to indicate constraint satisfaction.

Human evaluation metric. Most of the aforementioned CTG-specific metrics for the evaluation of text semantics and structures require the introduction of an additional semantic-relevant classifier or structure-relevant parsing tools, which may not be as reliable as the well-educated human participants. Therefore, human evaluation is also essential to test how a CTG model satisfies the control elements. Similar to the Human-centric method described in Section 4.1.1, human assessors are always directly asked to score the relevance of the generated texts to the control attributes (e.g., sentiment, topic, toxic, contextual relevancy, and common sense [10, 14, 24, 74, 151, 158]), and the average score obtained is used as the final evaluation result.

4.3 Summary

Human-centric evaluation is the most precise method for evaluating the quality of system-generated texts. It would be ideal when human resources permit, which unfortunately is not always the case. Thus, this type of evaluation suffers from various weaknesses: (1) **Expensive and time-consuming** — the recruitment and selection of evaluators, the setup of the evaluation and other steps all require time and manpower, especially for the evaluation tasks that can only be handled by domain experts; (2) **Maintaining quality control** [52, 90] — although the emergence of online crowd-sourcing platforms has eased the cost problem to a certain extent, the quality of personnel conducting online evaluations cannot always be guaranteed; (3) **Lack of consistency** [136] — the reproducibility of the evaluation results can be low due to the change of evaluation personnel, which leads to the inconsistency problem.

Compared with human-centric evaluation metrics, automatic evaluation metrics are easy to use, obtain results quickly, and are low cost. However, the evaluation of this type is less precise than human assessments. It is vital in the future to develop automated evaluation methods comparable to the human level of evaluation. The semi-automatic evaluation metric combines the advantages of human-centric methods and automatic methods, but still requires a lot of human judgments or labeling, which are expensive and time-consuming. One key future research direction is to

develop a better way to augment human judgments with automatic evaluation, or vice versa, to obtain improved evaluation quality and diversity.

When it comes to the CTG tasks, we not only need to use the general NLG evaluation metrics but also need specific metrics to evaluate whether the generated texts are consistent with the controlled elements. Generally speaking, the consistency of controlled elements and generated texts are relatively easy to measure in most cases. The main challenges of text quality evaluation still lie in the general evaluation methods.

5 CHALLENGES AND FUTURE DIRECTIONS

5.1 Challenges

The PLMs have mastered a remarkable level of linguistic knowledge (semantic, syntax, etc.) from large-scale corpus, naturally enabling the production of more fluent and diverse text. However, due to the black box characteristics of neural networks, the general PLMs are still not sufficiently controllable during the text generation process. How to fully exploit the powerful PLMs to generate the desired and controllable text has become a promising yet challenging field in both academia and industry. Based on the systematic review of the key concepts, methods, and findings in the latest developments in PLM-based controllable text generation, we think this promising and fast-growing area is still facing a number of challenges.

First, PLMs have learned rich knowledge from large-scale corpus used for pre-training. However, an NLG model needs to learn control constraints on its own training corpus. It is often difficult for the existing PLM-based models to ensure the domain diversity of the generated text while pursuing controllability. This is indeed the well-known catastrophic forgetting problem in PLM. In the field of text generation, it is still a challenge to overcome this problem and improve the ability of the PLM-based NLG model to generate multi-domain text that satisfies specific control conditions, with few or zero domain-specific samples.

Second, controlling the generation of text in the decoding stage of a generative model is a low-cost method of model training. It can maintain the characteristics of the original language model to the greatest extent. However, in most cases, the existing methods are relatively rudimentary, and only use the external decoupled attribute discriminator to control the attributes. There is a distribution gap between the discriminator and the generator, leading to a coarser granularity in the guidance process and decreased quality of the generated text. In addition, it is difficult to directly apply the decoding-time approachesto fine-grained control scenarios such as data-to-text or multi-attribute control tasks.

Third, from the perspective of probability theory, a generative pre-trained language model (referring specifically to the GPT-like models) is essentially an enhanced version of dense conditional probability $p(x_n \mid x_1, x_2, \ldots, x_{n-1})$ to describe the probability distribution of natural language. However, this local normalization format has certain limitations in paragraph/document-level modeling. For example, it is hard to keep long-range coherence in terms of both semantic logic and controlled condition. It calls for further research to establish a global normalization based on PLMs to ensure that text generation can be controlled locally and globally at the same time.

Fourth, the construction of large-scale PLMs is typically data driven, which allows the models to learn the primary logic and commonsense knowledge contained in the training corpus. However, the knowledge captured in those models is often rather superficial. The PLMs will lose generalization ability when the training data does not contain relevant commonsense and domain-specific knowledge. Therefore, purely relying on PLMs could be difficult to control the generated texts faithfully with respect to common sense and rich knowledge specific to the target domain.

Fifth, reasonable and reliable evaluation has always been a bottleneck restricting the development of more advanced text generation technologies. This is also the case for controllable text

generation. Generally speaking, the satisfaction of controlled conditions is relatively easy to evaluate. However, there is still a lack of an objective, accurate, and comprehensive evaluation mechanism that is fully compatible with human judgment. For controllable text generation, in addition to the control conditions, the quality of the text itself is equally important. If the quality of the generated text of an NLG model cannot be accurately evaluated, it is hard to think of a way to control it.

Finally, we believe that the research on controlled text generation is still in its early stage. In Section 2.3 of this article, we have summarized a range of tasks involving CTG. However, few of them are actually dedicated CTG tasks. With the rapid development of text generation, there is a need to come up with dedicated benchmarking tasks and datasets for CTG with diverse control requirements.

5.2 Future Directions

Based on the summary of current work and the challenges mentioned above, we suggest the following promising future directions for PLM-based controllable text generation.

Prompt-based Learning: Prompt-based learning has become a new way for fine-tuning PLMs. Based on the well-designed prompting function, a PLM is able to perform few-shot or even zero-shot learning, adapting to new scenarios with few or no labeled data, thus overcoming the problem of catastrophic forgetting. The above features are also attractive for controlled text generation, since the prompt-based methods are able to generate more diverse text fields and increase the distribution space of the text to be filtered, so that it is theoretically more possible to produce text that meets the specific control conditions. Currently, the application of prompt-based methods and their variants (e.g., in-context learning [148], instruction tuning [94], etc.) in large-scale language models is a hot academic topic. There is also great potential in finding better ways to apply this paradigm to controllable text generation.

Fine-grained Decoding Control: More fine-grained decoding control methods need to be explored. On the one hand, the decoding-time methods can be improved to achieve more effective control. For example, co-training between the guided model and the generative model ensures finer-grained text generation. On the other hand, the existing single-attribute (e.g., emotion, topic, etc.) controlled tasks can also be extended to multi-attribute controlled tasks in a unified framework, achieving the simultaneous control of multiple aspects for a generated sentence.

Integration with Classic Generative Theory and Linguistic Knowledge: The PLM-based controllable text generation task can be regarded as obtaining a natural language distribution constructed by a PLM, the corresponding distribution that satisfies certain specified constraints. This process is intrinsically related to classic generative models such as Generative Adversarial Networks (GANs) [41], Variational Autoencoders (VAEs) [60], Energy-based Models [25] and their variants. It is known that auto-regressive PLMs, i.e., the GPT family models, cannot model global information of the generated texts naturally, making it difficult to control the generated text's distribution at the paragraph/document level. We expect that combining classic probability theory to bridge the gap between PLMs and traditional generative models will help solve the problem at the theoretical level.

In addition, auto-regressive PLM is essentially a locally normalized method, allowing it to produce fluent text in short text-generation scenarios. From a linguistic point of view, it is believed that more linguistic knowledge, such as paragraph/document level structures and logic, are needed in long text generation, which PLMs cannot provide directly. A promising solution is to combine linguistic knowledge with PLMs to overcome the inherent problems of auto-regressive models in long text modeling to better ensure quality and controllability.

Incorporation of External Knowledge: Introducing additional knowledge to enhance the PLM's generation is a promising direction. One direct idea is to combine with information retrieval

and allow the PLM to refer to the retrieved information from the Web or domain data repositories, ensuring that the generated content is more reasonable and alleviating the problem of hallucinations [53]. Furthermore, a knowledge graph is a natural carrier of explicit domain-specific knowledge that also provides effective reasoning mechanisms. Therefore, it can be a good complement to the use of PLMs, which lack domain-specific knowledge and logical reasoning capabilities.

Novel Evaluation Metrics and Methods: Developing innovative evaluation metrics for CTG is still an important topic that needs to be further studied from both the general text generation perspective (such as fluency, diversity, and coherence) and the CTG-specific perspective (such as fidelity). Since the pre-trained language models have mastered a great deal of semantic and grammatical knowledge, applying them in reverse to assess the text quality of an NLG model would be an interesting and fascinating area for further investigation.

New CTG tasks: Controllable text generation is a broad concept that has gained significant attention in recent years. With the rapid development of large-scale language models, such as ChatGPT and GPT-4, there is a growing interest in exploring new standards and tasks that align with the goal of achieving Artificial General Intelligence (AGI). One promising direction is to define AGI-oriented benchmarks and tasks that aim to control the language model to produce accurate and reliable information or ensure that the generated content is aligned with human values and does not have the harmful effects.

6 CONCLUSIONS

In this article, we have comprehensively summarized the typical applications, main approaches, and evaluation methodologies of controllable text generation based on large-scale pre-trained language models. Based on the critical analysis of the existing methods, we have identified a series of key challenges in this field and highlighted several promising future directions. Large-scale pre-trained language models have brought unprecedented opportunities for the development of controllable text generation technologies, calling for more researchers to join the field and create a new era of exploration of the field. We are hopeful that this literature survey is able to provide a clear picture of the field and set a roadmap for researchers and practitioners to move forward.

7 ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] Ali Amin-Nejad, Julia Ive, and Sumithra Velupillai. 2020. Exploring Transformer text generation for medical dataset augmentation. In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 4699–4708. https://aclanthology.org/2020.lrec-1.578

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, 936–945. https://doi.org/10.18653/v1/d17-1098

[3] Wilker Aziz, Sheila Castilho, and Lucia Specia. 2012. PET: A tool for post-editing and assessing machine translation. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), Istanbul, Turkey, 3982–3987. http://www.lrec-conf.org/proceedings/lrec2012/pdf/985_Paper.pdf

[4] Soumya Barikeri, Anne Lauscher, Ivan Vulić, and Goran Glavaš. 2021. RedditBias: A real-world resource for bias evaluation and debiasing of conversational language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 1941–1955. https://doi.org/10.18653/v1/2021.acl-long.151

[5] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document Transformer. *CoRR* abs/2004.05150 (2020). arXiv:2004.05150. https://arxiv.org/abs/2004.05150

[6] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT'21)* (Virtual Event, Canada). Association for Computing Machinery, New York, NY, USA, 610–623. https://doi.org/10.1145/3442188.3445922

[7] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.* 3, null (Mar 2003), 1137–1155. https://dl.acm.org/doi/pdf/10.5555/944919.944966

[8] Sumanta Bhattacharyya, Amirmohammad Rooshenas, Subhajit Naskar, Simeng Sun, Mohit Iyyer, and Andrew McCallum. 2021. Energy-based reranking: Improving neural machine translation using energy-based models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 4528–4537. https://doi.org/10.18653/v1/2021.acl-long.349

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[10] Fredrik Carlsson, Joey Öhman, Fangyu Liu, Severine Verlinden, Joakim Nivre, and Magnus Sahlgren. 2022. Fine-grained controllable text generation using non-residual prompting. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 6837–6857. https://doi.org/10.18653/v1/2022.acl-long.471

[11] Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 552–562. https://doi.org/10.18653/v1/D19-1052

[12] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1662–1675. https://doi.org/10.18653/v1/N18-1150

[13] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *CoRR* abs/2006.14799 (2020). arXiv:2006.14799. https://arxiv.org/abs/2006.14799

[14] Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2021. CoCon: A self-supervised approach for controlled text generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=VD_ozqvBy4W

[15] Khyathi Chandu, Shrimai Prabhumoye, Ruslan Salakhutdinov, and Alan W. Black. 2019. "My way of telling a story": Persona based grounded story generation. In *Proceedings of the 2nd Workshop on Storytelling*. Association for Computational Linguistics, Florence, Italy, 11–21. https://doi.org/10.18653/v1/W19-3402

[16] Haw-Shiuan Chang, Jiaming Yuan, Mohit Iyyer, and Andrew McCallum. 2021. Changing the mind of Transformers for topically-controllable language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 2601–2611. https://doi.org/10.18653/v1/2021.eacl-main.223

[17] Cheng-Kuan Chen, Zhufeng Pan, Ming-Yu Liu, and Min Sun. 2019. Unsupervised stylish image description generation via domain layer norm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8151–8158.

[18] Huimin Chen, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, and Zhipeng Guo. 2019. Sentiment-controllable Chinese poetry generation. In *Proceedings of the T28th International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4925–4931. https://doi.org/10.24963/ijcai.2019/684

[19] 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* (2022).

[20] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

[21] Alexis Conneau and Guillaume Lample. 2019. *Cross-Lingual Language Model Pretraining*. Curran Associates Inc., Red Hook, NY, USA. https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf

[22] Daniel Dahlmeier, Chang Liu, and Hwee Tou Ng. 2011. TESLA at WMT 2011: Translation evaluation and tunable metric. In *Proceedings of the 6th Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Edinburgh, Scotland, 78–84. https://aclanthology.org/W11-2106

[23] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2978–2988. https://doi.org/10.18653/v1/P19-1285

[24] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1edEyBKDS

[25] Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 2020. Residual energy-based models for text generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1l4SgHKDH

[26] Michael Denkowski, Chris Dyer, and Alon Lavie. 2014. Learning from post-editing: Online model adaptation for statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Gothenburg, Sweden, 395–404. https://doi.org/10.3115/v1/E14-1042

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[28] Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2020. Queens are powerful too: Mitigating gender bias in dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 8173–8188. https://doi.org/10.18653/v1/2020.emnlp-main.656

[29] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. *Unified Language Model Pre-Training for Natural Language Understanding and Generation*. Curran Associates Inc., Red Hook, NY, USA. https://dl.acm.org/doi/pdf/10.5555/3454287.3455457

[30] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. 2021. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. *CoRR* abs/2103.03404 (2021). arXiv:2103.03404 https://arxiv.org/abs/2103.03404

[31] Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1342–1352. https://doi.org/10.18653/v1/P17-1123

[32] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 889–898. https://doi.org/10.18653/v1/P18-1082

[33] Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Outline to story: Fine-grained controllable story generation from cascaded events. *arXiv preprint arXiv:2101.00822* (2021). https://arxiv.org/pdf/2101.00822v1.pdf

[34] Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*. Association for Computational Linguistics, Copenhagen, Denmark, 94–104. https://doi.org/10.18653/v1/W17-4912

[35] Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. 2022. EmoSen: Generating sentiment and emotion controlled responses in a multimodal dialogue system. *IEEE Transactions on Affective Computing* 13, 3 (2022), 1555–1566. https://doi.org/10.1109/TAFFC.2020.3015491

[36] Mauajama Firdaus, Arunav Shandilya, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Being polite: Modeling politeness variation in a personalized dialog agent. *IEEE Transactions on Computational Social Systems* (2022), 1–10. https://doi.org/10.1109/TCSS.2022.3182986

[37] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 3816–3830. https://doi.org/10.18653/v1/2021.acl-long.295

[38] Xiang Gao, Yizhe Zhang, Sungjin Lee, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2019. Structuring latent spaces for stylized response generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 1814–1823. https://doi.org/10.18653/v1/D19-1190

[39] Dimitra Gkatzia and Saad Mahamood. 2015. A snapshot of NLG evaluation practices 2005–2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*. Association for Computational Linguistics, Brighton, UK, 57–60. https://doi.org/10.18653/v1/W15-4708

[40] Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with Aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 4319–4338. https://doi.org/10.18653/v1/2020.emnlp-main.351

[41] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[42] Aditya Grover, Jiaming Song, Alekh Agarwal, Kenneth Tran, Ashish Kapoor, Eric Horvitz, and Stefano Ermon. 2019. *Bias Correction of Learned Generative Models Using Likelihood-Free Importance Weighting*. Curran Associates Inc., Red Hook, NY, USA. https://papers.nips.cc/paper/2019/file/d76d8deea9c19cc9aaf2237d2bf2f785-Paper.pdf

[43] Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 1689–1701. https://doi.org/10.18653/v1/N19-1169

[44] Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. *ArXiv* abs/2002.03912 (2020).

[45] Xingwei He. 2021. Parallel refinements for lexically constrained text generation with BART. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 8653–8666. https://doi.org/10.18653/v1/2021.emnlp-main.681

[46] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rygGQyrFvH

[47] Helmut Horacek. 2001. Building natural language generation systems - Ehud Reiter and Robert Dale (Eds.), University of Aberdeen and Macquarie University, Cambridge University Press, 2000, ISBN 0-521-62036-8. *Artif. Intell. Medicine* 22, 3 (2001), 277–280. https://doi.org/10.1016/S0933-3657(00)00114-7

[48] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Controllable text generation. *CoRR* abs/1703.00955 (2017). arXiv:1703.00955. http://arxiv.org/abs/1703.00955

[49] Xinyu Hua and Lu Wang. 2020. PAIR: Planning and iterative refinement in pre-trained Transformers for long text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 781–793. https://doi.org/10.18653/v1/2020.emnlp-main.57

[50] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)* 38, 3 (2020), 1–32.

[51] Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2020. Reducing sentiment bias in language models via counterfactual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 65–83. https://doi.org/10.18653/v1/2020.findings-emnlp.7

[52] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP'10)* (Washington DC). Association for Computing Machinery, New York, NY, USA, 64–67. https://doi.org/10.1145/1837885.1837906

[53] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.* 55, 12, Article 248 (mar 2023), 38 pages. https://doi.org/10.1145/3571730

[54] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics* 8 (2020), 423–438. https://doi.org/10.1162/tacl_a_00324

[55] Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6505–6520. https://doi.org/10.18653/v1/2020.emnlp-main.527

[56] Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*. Association for Computational Linguistics, Dublin, Ireland, 97–102. https://aclanthology.org/2020.inlg-1.14

[57] Pei Ke, Hao Zhou, Yankai Lin, Peng Li, Jie Zhou, Xiaoyan Zhu, and Minlie Huang. 2022. CTRLEval: An unsupervised reference-free metric for evaluating controlled text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2306–2319. https://doi.org/10.18653/v1/2022.acl-long.164

[58] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional Transformer language model for controllable generation. *CoRR* abs/1909.05858 (2019). arXiv:1909.05858. http://arxiv.org/abs/1909.05858

[59] Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2021. A distributional approach to controlled text generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=jWkw45-9AbL
[60] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).
[61] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient Transformer. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkgNKkHtvB
[62] Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 4929–4952. https://doi.org/10.18653/v1/2021.findings-emnlp.424
[63] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 3045–3059. https://doi.org/10.18653/v1/2021.emnlp-main.243
[64] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 110–119. https://doi.org/10.18653/v1/N16-1014
[65] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 994–1003. https://doi.org/10.18653/v1/P16-1094
[66] Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 742–751. https://doi.org/10.18653/v1/2020.acl-main.68
[67] Shifeng Li, Shi Feng, Daling Wang, Kaisong Song, Yifei Zhang, and Weichao Wang. 2020. EmoElicitor: An open domain response generation model with user emotional reaction awareness. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3637–3643. https://doi.org/10.24963/ijcai.2020/503. Main track.
[68] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 4582–4597. https://doi.org/10.18653/v1/2021.acl-long.353
[69] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. 2022. Diffusion-LM improves controllable text generation. *ArXiv* abs/2205.14217 (2022). https://arxiv.org/abs/2205.14217
[70] Yi Liao, Yasheng Wang, Qun Liu, and Xin Jiang. 2019. GPT-based generation for classical Chinese poetry. *arXiv preprint arXiv:1907.00151* (2019). https://arxiv.org/abs/1907.00151
[71] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013
[72] Zhaojiang Lin, Andrea Madotto, Yejin Bang, and Pascale Fung. 2021. The adapter-bot: All-in-one controllable conversational model. In *35th AAAI Conference on Artificial Intelligence, AAAI 2021, 33rd Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The 11th Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 16081–16083. https://ojs.aaai.org/index.php/AAAI/article/view/18018
[73] Zhiyu Lin and Mark O. Riedl. 2021. Plug-and-blend: A framework for plug-and-play controllable story generation with sketches. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 17. 58–65. https://arxiv.org/abs/2104.04039
[74] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 6691–6706. https://doi.org/10.18653/v1/2021.acl-long.522
[75] Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2010. TESLA: Translation evaluation of sentences with linear-programming-based analysis. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR*. Association for Computational Linguistics, Uppsala, Sweden, 354–359. https://aclanthology.org/W10-1754
[76] Danyang Liu, Juntao Li, Meng-Hsuan Yu, Ziming Huang, Gongshen Liu, Dongyan Zhao, and Rui Yan. 2020. A character-centric neural model for automated story generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1725–1732. https://ojs.aaai.org//index.php/AAAI/article/view/5536

[77] 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. *CoRR* abs/2107.13586 (2021). arXiv:2107.13586 https://arxiv.org/abs/2107.13586

[78] Ruibo Liu, Chenyan Jia, Jason Wei, Guangxuan Xu, Lili Wang, and Soroush Vosoughi. 2021. Mitigating political bias in language models through reinforced calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 14857–14866. https://arxiv.org/abs/2104.14795

[79] Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020. Data boost: Text data augmentation through reinforcement learning guided conditional generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 9031–9041. https://doi.org/10.18653/v1/2020.emnlp-main.726

[80] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics* 8 (2020), 726–742. https://doi.org/10.1162/tacl_a_00343

[81] Ye Liu, Wolfgang Maier, Wolfgang Minker, and Stefan Ultes. 2021. Naturalness evaluation of natural language generation in task-oriented dialogues using BERT. *arXiv preprint arXiv:2109.02938* (2021).

[82] 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. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692. http://arxiv.org/abs/1907.11692

[83] Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/7cf64379eb6f29a4d25c4b6a2df713e4-Paper.pdf

[84] Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic Turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1116–1126. https://doi.org/10.18653/v1/P17-1103

[85] Fuli Luo, Damai Dai, Pengcheng Yang, Tianyu Liu, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Learning to control the fine-grained sentiment for story ending generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 6020–6026. https://doi.org/10.18653/v1/P19-1603

[86] Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*. Association for Computational Linguistics, Hong Kong, 90–98. https://doi.org/10.18653/v1/D19-5609

[87] Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2799–2808. https://doi.org/10.18653/v1/P19-1269

[88] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[89] Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. 2022. Mix and match: Learning-free controllable text generation using energy language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 401–415. https://doi.org/10.18653/v1/2022.acl-long.31

[90] Tanushree Mitra, C. J. Hutto, and Eric Gilbert. 2015. Comparing person- and process-centric strategies for obtaining quality data on Amazon Mechanical Turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)* (Seoul, Republic of Korea). Association for Computing Machinery, New York, NY, USA, 1345–1354. https://doi.org/10.1145/2702123.2702553

[91] Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 432–447. https://doi.org/10.18653/v1/2021.naacl-main.37

[92] Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics* 6 (2018), 373–389. https://transacl.org/ojs/index.php/tacl/article/view/1424

[93] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. RankME: Reliable human ratings for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 72–78. https://doi.org/10.18653/v1/N18-2012

[94] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 27730–27744. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

[95] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[96] Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 3973–3997. https://aclanthology.org/2021.findings-emnlp.334

[97] Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 172–182. https://doi.org/10.18653/v1/2020.findings-emnlp.17

[98] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[99] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Yejin Choi, and Zaïd Harchaoui. 2021. MAUVE: Human-machine divergence curves for evaluating open-ended text generation. *CoRR* abs/2102.01454 (2021). arXiv:2102.01454. https://arxiv.org/abs/2102.01454

[100] Shrimai Prabhumoye, Alan W. Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 1–14. https://doi.org/10.18653/v1/2020.coling-main.1

[101] Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 480–489.

[102] Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 6908–6915.

[103] Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 2912–2924. https://doi.org/10.18653/v1/2022.findings-acl.229

[104] Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. 2019. Reducing gender bias in word-level language models with a gender-equalizing loss function. *arXiv preprint arXiv:1905.12801* (2019).

[105] Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. 2019. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Márquez (Eds.). Association for Computational Linguistics, 5427–5436. https://www.aclweb.org/anthology/P19-1539/

[106] Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. COLD Decoding: Energy-based Constrained Text Generation with Langevin Dynamics. https://doi.org/10.48550/ARXIV.2202.11705

[107] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[108] 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 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[109] Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021. Increasing faithfulness in knowledge-grounded dialogue with controllable features. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 704–718. https://doi.org/10.18653/v1/2021.acl-long.58

[110] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[111] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*. Association for Computational Linguistics, Online, 211–227. https://doi.org/10.18653/v1/2021.nlp4convai-1.20

[112] Leonardo F. R. Ribeiro, Yue Zhang, and Iryna Gurevych. 2021. Structural adapters in pretrained language models for AMR-to-text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 4269–4282. https://aclanthology.org/2021.emnlp-main.351

[113] Yu-Ping Ruan and Zhenhua Ling. 2021. Emotion-regularized conditional variational autoencoder for emotional response generation. *IEEE Transactions on Affective Computing* (2021). https://arxiv.org/abs/2104.08857

[114] Punyajoy Saha, Kanishk Singh, Adarsh Kumar, Binny Mathew, and Animesh Mukherjee. 2022. CounterGeDi: A controllable approach to generate polite, detoxified and emotional counterspeech. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 5157–5163. https://doi.org/10.24963/ijcai.2022/716. AI for Good.

[115] Bidisha Samanta, Mohit Agarwal, and Niloy Ganguly. 2020. Fine-grained sentiment Controlled text generation. *arXiv preprint arXiv:2006.09891* (2020). https://arxiv.org/abs/2006.09891

[116] 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* (2022).

[117] Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. Discriminative adversarial search for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 793, 10 pages. https://arxiv.org/abs/2002.10375

[118] Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7881–7892. https://doi.org/10.18653/v1/2020.acl-main.704

[119] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 35–40. https://doi.org/10.18653/v1/N16-1005

[120] Yizhan Shao, Tong Shao, Minghao Wang, Peng Wang, and Jie Gao. 2021. A sentiment and style controllable approach for chinese poetry generation. In *CIKM'21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 4784–4788. https://doi.org/10.1145/3459637.3481964

[121] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2020. Towards controllable biases in language generation. *arXiv preprint arXiv:2005.00268* (2020).

[122] Zhonghao Sheng, Kaitao Song, Xu Tan, Yi Ren, Wei Ye, Shikun Zhang, and Tao Qin. 2021. SongMASS: Automatic song writing with pre-training and alignment constraint. In *35th AAAI Conference on Artificial Intelligence, AAAI 2021, 33rd Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The 11th Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 13798–13805. https://ojs.aaai.org/index.php/AAAI/article/view/17626

[123] Wenxian Shi, Hao Zhou, Ning Miao, and Lei Li. 2020. Dispersed exponential family mixture VAEs for interpretable text generation. In *International Conference on Machine Learning*. PMLR, 8840–8851. http://proceedings.mlr.press/v119/shi20f/shi20f.pdf

[124] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 4222–4235. https://doi.org/10.18653/v1/2020.emnlp-main.346

[125] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. 2015. Learning structured output representation using deep conditional generative models. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'15)* (Montreal, Canada). MIT Press, Cambridge, MA, USA, 3483–3491. https://dl.acm.org/doi/10.5555/2969442.2969628

[126] Haoyu Song, Yan Wang, Kaiyan Zhang, Wei-Nan Zhang, and Ting Liu. 2021. BoB: BERT over BERT for training persona-based dialogue models from limited personalized data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021 (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 167–177. https://doi.org/10.18653/v1/2021.acl-long.14

[127] Haoyu Song, Yan Wang, Weinan Zhang, Xiaojiang Liu, and Ting Liu. 2020. Generate, delete and rewrite: A three-stage framework for improving persona consistency of dialogue generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 5821–5831. https://doi.org/10.18653/v1/2020.acl-main.516

[128] Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020. Structural information preserving for graph-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 7987–7998. https://doi.org/10.18653/v1/2020.acl-main.712

[129] Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuanjing Huang. 2019. Generating responses with a specific emotion in dialog. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 3685–3695. https://doi.org/10.18653/v1/p19-1359

[130] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)* (Vancouver, BC, Canada). Curran Associates Inc., Red Hook, NY, USA, Article 253, 14 pages. https://proceedings.neurips.cc/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf

[131] Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 895–909. https://aclanthology.org/2021.findings-emnlp.76

[132] Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J. Martin, Animesh Mehta, Brent Harrison, and Mark O. Riedl. 2019. Controllable neural story plot generation via reward shaping. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 5982–5988. https://doi.org/10.24963/ijcai.2019/829

[133] Hongyin Tang, Miao Li, and Beihong Jin. 2019. A topic augmented text generation model: Joint learning of semantics and structural features. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 5090–5099. https://doi.org/10.18653/v1/D19-1513

[134] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[135] Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. 2020. ENGINE: Energy-based inference networks for non-autoregressive machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2819–2826. https://doi.org/10.18653/v1/2020.acl-main.251

[136] Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language* 67 (2021), 101151. https://doi.org/10.1016/j.csl.2020.101151

[137] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008. https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[138] Olga Vechtomova, Hareesh Bahuleyan, Amirpasha Ghabussi, and Vineet John. 2018. Generating lyrics with variational autoencoder and multi-modal artist embeddings. *CoRR* abs/1812.08318 (2018). arXiv:1812.08318. http://arxiv.org/abs/1812.08318

[139] Dingmin Wang, Ziyao Chen, Wanwei He, Li Zhong, Yunzhe Tao, and Min Yang. 2021. A template-guided hybrid pointer network for knowledge-based task-oriented dialogue systems. In *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*. Association for Computational Linguistics, Online, 18–28. https://doi.org/10.18653/v1/2021.dialdoc-1.3

[140] Ke Wang and Xiaojun Wan. 2018. SentiGAN: Generating sentimental texts via mixture adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 4446–4452. https://doi.org/10.24963/ijcai.2018/618

[141] Ruize Wang, Zhongyu Wei, Ying Cheng, Piji Li, Haijun Shan, Ji Zhang, Qi Zhang, and Xuanjing Huang. 2020. Keep it consistent: Topic-aware storytelling from an image stream via iterative multi-agent communication. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 2250–2260. https://doi.org/10.18653/v1/2020.coling-main.204

[142] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational auto-encoder for text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 166–177. https://doi.org/10.18653/v1/N19-1015

[143] Yufei Wang, Ian Wood, Stephen Wan, Mark Dras, and Mark Johnson. 2021. Mention flags (MF): Constraining Transformer-based text generators. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 103–113. https://doi.org/10.18653/v1/2021.acl-long.9

[144] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*. https://openreview.net/forum?id=gEZrGCozdqR

[145] Wei Wei, Jiayi Liu, Xianling Mao, Guibing Guo, Feida Zhu, Pan Zhou, and Yuchong Hu. 2019. Emotion-aware chat machine: Automatic emotional response generation for human-like emotional interaction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1401–1410. https://doi.org/10.1145/3357384.3357937

[146] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*. https://arxiv.org/abs/1908.04319

[147] Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, and Bill Dolan. 2021. A controllable model of grounded response generation. In *35th AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 14085–14093. https://ojs.aaai.org/index.php/AAAI/article/view/17658

[148] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*. https://openreview.net/forum?id=RdJVFCHjUMI

[149] Peng Xu, Yanshuai Cao, and Jackie Chi Kit Cheung. 2019. Unsupervised controllable text generation with global variation discovery and disentanglement. *CoRR* abs/1905.11975 (2019). arXiv:1905.11975. http://arxiv.org/abs/1905.11975

[150] Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 2831–2845. https://doi.org/10.18653/v1/2020.emnlp-main.226

[151] Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 3511–3535. https://doi.org/10.18653/v1/2021.naacl-main.276

[152] Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2022. Tailor: A Prompt-Based Approach to Attribute-Based Controlled Text Generation. https://doi.org/10.48550/ARXIV.2204.13362

[153] Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. 2019. Enhancing topic-to-essay generation with external commonsense knowledge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2002–2012. https://doi.org/10.18653/v1/P19-1193

[154] Xiaopeng Yang, Xiaowen Lin, Shunda Suo, and Ming Li. 2018. Generating thematic Chinese poetry using conditional variational autoencoders with hybrid decoders. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)* (Stockholm, Sweden). AAAI Press, 4539–4545. https://arxiv.org/abs/1711.07632

[155] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf

[156] Y. Zeldes, D. Padnos, O. Sharir, and B. Peleg. 2020. Technical report: Auxiliary tuning and its application to conditional text generation. (2020). https://arxiv.org/abs/2006.16823

[157] Yan Zeng and Jian-Yun Nie. 2021. A simple and efficient multi-task learning approach for conditioned dialogue generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 4927–4939. https://doi.org/10.18653/v1/2021.naacl-main.392

[158] Hanqing Zhang and Dawei Song. 2022. DisCup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 3392–3406. https://aclanthology.org/2022.emnlp-main.223

[159] Rui Zhang, Zhenyu Wang, Kai Yin, and Zhenhua Huang. 2019. Emotional text generation based on cross-domain sentiment transfer. *IEEE Access* 7 (2019), 100081–100089. https://ieeexplore.ieee.org/document/8772090

[160] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too?. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 2204–2213. https://doi.org/10.18653/v1/P18-1205

[161] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=SkeHuCVFDr

[162] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Online, 270–278. https://doi.org/10.18653/v1/2020.acl-demos.30

[163] Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. 2020. POINTER: Constrained progressive text generation via insertion-based generative pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 8649–8670. https://doi.org/10.18653/v1/2020.emnlp-main.698

[164] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1441–1451. https://doi.org/10.18653/v1/P19-1139

[165] Chao Zhao, Marilyn A. Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2481–2491. https://doi.org/10.18653/v1/2020.acl-main.224

[166] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. 2017. Energy-based generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=ryh9pmcee

[167] Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. 2020. A pre-training based personalized dialogue generation model with persona-sparse data. In *The 34th AAAI Conference on Artificial Intelligence, AAAI 2020, The 32nd Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The 10th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 9693–9700. https://aaai.org/ojs/index.php/AAAI/article/view/6518

[168] Peixiang Zhong, Chen Zhang, Hao Wang, Yong Liu, and Chunyan Miao. 2020. Towards persona-based empathetic conversational models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6556–6566. https://doi.org/10.18653/v1/2020.emnlp-main.531

[169] Wangchunshu Zhou and Ke Xu. 2020. Learning to compare for better training and evaluation of open domain natural language generation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9717–9724. https://arxiv.org/abs/2002.05058

[170] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'18)* (Ann Arbor, MI, USA). Association for Computing Machinery, New York, NY, USA, 1097–1100. https://doi.org/10.1145/3209978.3210080

[171] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *CoRR* abs/1909.08593 (2019). arXiv:1909.08593. http://arxiv.org/abs/1909.08593

[172] Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. Controllable generation from pre-trained language models via inverse prompting. *CoRR* abs/2103.10685 (2021). arXiv:2103.10685. https://arxiv.org/abs/2103.10685

Received 10 January 2022; revised 14 July 2023; accepted 21 August 2023