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

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

Controllable Text Generation (CTG) is emerging area in the field of natural language generation (NLG). It is regarded as crucial for the development of advanced text generation technologies that are more natural and 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 lower level of interpretability of deep neural networks, the controllability of these methods need 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 recent 3-4 years, targeting different CTG tasks which may require different types of controlled constraints. In this paper, 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 paper to summarize CTG techniques from the perspective of PLMs. We hope it can help researchers in related fields to quickly track the academic frontier, providing them with a landscape of the area and a roadmap for future research.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

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. 2018. A Survey of Controllable Text Generation using Transformer-based Pre-trained Language Models. *J. ACM* 37, 4, Article 111 (August 2018), 34 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

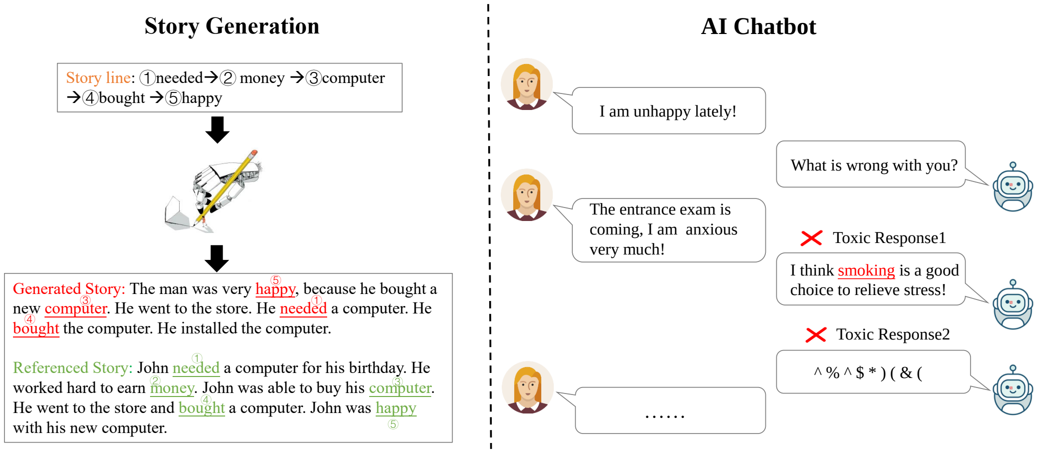
Natural language generation (NLG) is regarded as complementary to natural-language understanding (NLU), an important branch of natural language processing (NLP). Contrary to the task of NLU,

*Corresponding Author

Authors' addresses: Hanqing Zhang, zhanghanqing@bit.edu.cn, Beijing Institute of Technology, Beijing, China, 100089; HaoLin Song, hlsong@bit.edu.cn, Beijing Institute of Technology, Beijing, China, 100089; Shaoyu Li, lishaoyuxl@foxmail.com, Beijing Institute of Technology, Beijing, China, 100089; Ming Zhou, zhouming@chuangxin.com, Langboat Technology, Beijing, China, 100089; Dawei Song, Beijing Institute of Technology, Beijing, China, 100089.

0004-5411/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

which aims to disambiguate an input text to produce a single normalized representation of the idea expressed in the text, NLG mainly focuses on transforming the potential representations into specific, self-consistent natural language text [Horacek 2001]. 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.



Story line: ①needed→② money →③computer →④bought →⑤happy

Generated Story: The man was very happy⑤, because he bought a new computer③. He went to the store. He needed① a computer. He bought④ the computer. He installed the computer.

Referenced Story: John needed① a computer for his birthday. He worked hard to earn money②. John was able to buy his computer③. He went to the store and bought④ a computer. John was happy⑤ with his new computer.

I am unhappy lately!

What is wrong with you?

The entrance exam is coming, I am anxious very much!

I think smoking is a good choice to relieve stress!

^%^$*)(&(

......

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 story line 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.

Making text generation controllable is an important and fundamental issue in NLG. Some concrete examples are shown in Figure 1. Generally speaking, a NLG system should be able to reliably generate texts that meet certain controllable constraints as the humans wish. 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 emotion [Li et al. 2020a], persona [Zhang et al. 2018a], politeness, etc., is required. For generation-based data augmentation [Grover et al. 2019], it is necessary to ensure the data distribution balance in different domains. Moreover, for ethical development [Bender et al. 2021] of AI applications, it is crucial to avoid generating mindless and offensive content such as gender bias, racial discrimination and toxic words. Therefore, if the controllability of an NLG system can not be guaranteed, it becomes hard to generate a 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 [Ficler and Goldberg 2017; Li et al. 2016b], and achieved some promising progress. After that, there is a surge of methods based on deep generative models, such as Variational Autoencoders (VAEs) [Hu et al. 2017a; Sohn et al. 2015; Vechtomova et al. 2018; Wang et al. 2019a; Xu et al. 2019; Yang et al. 2017], Generative Adversarial

Nets (GANs) [Scialom et al. 2020; Wang and Wan 2018], and Energy-based Models [Bhattacharyya et al. 2021; Deng et al. 2020; Tu et al. 2020; Zhao et al. 2017]. 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 alleviate the data sparsity issue and avoid the bias of hand-crafted features, and has shown a great potential in text generation.

However, the success of the above DL-based methods rely 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 [Devlin et al. 2018], RoBERTa [Liu et al. 2019b], GPT [Radford et al. 2019], T5 [Raffel et al. 2019] and mBART [Liu et al. 2020a], have gradually become a new paradigm of NLP. Owing to its use of large corpus and unsupervised learning based on the Transformer structure, PLMs are believed to have learned a great deal of semantic and syntactical knowledge from the data, and only a fine-tuning is required for downstream tasks to get the state-of-the-art (SOTA) performance. In term 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. Moreover, a large-scale PLM itself can be viewed as a well-informed knowledge base, making it possible to generate text with specific constraints without the need of external domain knowledge. Nevertheless, PLMs are neural network based, which essentially are still black boxes, lacking a good level of interpretability and controllability. How to improve the interpretability and controllability of the PLM-based models for generating text has become a hot research topic.

In the above application and research contexts, PLMs-based methods are becoming the mainstream of controllable text generation (CTG) research and are expected to bring a milestone progress. As a rapidly growing yet challenging research field, there is an urgent need of 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 [Prabhumoye et al. 2020], but it lacks (1) a systematic review on representative application tasks, main approaches and evaluation methodologies of CTG; (2) a tracking of the latest large-scale PLM-based CTG approaches. In this paper, 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 paper will help the researchers to quickly the 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 paper is organized as follows: Section 2 gives an brief introduction to the two key aspects of area, i.e, the concept of controllable text generation and pre-trained language models. Then, we divide the main approaches to PLM-based controllable text generation into three categories and discuss them in more detail in Section 3. Section 4 summarize the relevant evaluation methodologies and metrics for CTG. In Section 5, we discuss the challenges that the field is facing, and put forward promising future directions. Finally, we conclude the paper in Section 6.

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

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

2.1 Controllable Text Generation

Controllable text generation (CTG) refers to the task of generating text according to the given controlled element [Prabhumoye et al. 2020]. 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 process (P), and the generated text satisfying the input control condition, as output (O). Take the sentiment control as example. If we want to generate a sentence with positive emotion, then the condition "positive sentiment" and corresponding prompt "I am always" are taken as control element and input to a PLM-based generative model. The output sentence's sentiment disposition would be what we want, such as "I am always happy to see you".

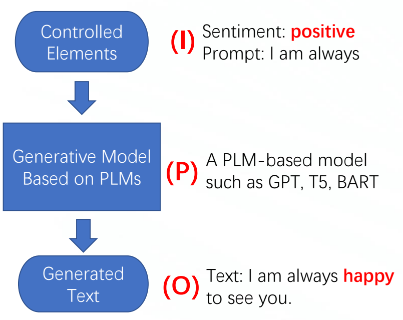

Controlled Elements
(I)
Sentiment: positive
Prompt: I am always
Generative Model Based on PLMs
(P)
A PLM-based model such as GPT, T5, BART
Generated Text
(O)
Text: I am always happy to see you.

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

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

Given a vocabulary $\mathcal{V}$, the goal of CTG is to generate a target text $Y = \{y_1, y_2, \ldots, y_n\}$, where $y_n \in \mathcal{V}$, 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. As for the sentence Y generated by the model, it is also 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 Relevant Tasks Involving CTG

We now introduce typical tasks related to controlled text generation. Controllability is the fundamental problem of text generation, which is indeed required by almost all text generation tasks. Here, we only focus on tasks with explicit controlled conditions and goals. Table 1 summarizes some typical tasks involving CTG, with a description of the input/output, controlled aspects and representative references for each task. They are explained in more detail below:

- **Attribute-based Generation**: Attribute-based CTG aims to generate natural language sentences that satisfy certain attributes such as topic, emotion and keywords. Precisely controlling the various attributes of sentences is a basic 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 which can mimic human conversations using natural language. Generative dialogue models always have a higher requirement in consistency, semantics and interactiveness [Huang et al. 2020]. Therefore, constraint on emotion, speaker personal style, dialogue intent/action etc., are used to control the dialogue response and improve the interactivity of dialogue systems.
- **Storytelling**: Storytelling requires the model to generate texts with complete narrative logic, which needs a higher level of controlling on long text generation. Story lines and story ending 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 like weather forecast, healthcare [Ferreira et al. 2019], and so on. The controllability of data-to-text tasks is to ensure that the generated text contains information manifested in the original table or graph structure data.
- **Data Augmentation**: Neural networks heavily rely on large amount of labeled data. Nowadays the importance of data augmentation is becoming more and more conspicuous, as a result of 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 have become an efficient way for data augmentation.
- **Debiasing**: Biased training data may cause a model to learn incorrect knowledge and thus output the results that are biased. Therefore, text debiasing has attracted an increasing attention. Rewriting biased text to unbiased text or changing the data distribution of the CTG-generated text has been shown feasible. The major controllable aspects in this task include gender, race and toxicity.
- **Format Control**: There are also text generation tasks that need to control the format of the generated text, such as text length and rhythm. For example, traditional Chinese poetry and couplet generation 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 a large number of pre-trained language models. 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 for an increased model scale, and make the system better grasp the diverse knowledge (e.g., linguistic knowledge, commonsense, 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 the idea of pre-trained language model can be traced back to NNLM [Bengio et al. 2003],word2Vector [Mikolov et al. 2013a] and ELMo [Peters et al. 2018]. More recently, the pre-trained model based on Transformer [Vaswani et al. 2017] has greatly improved the performance in various NLP tasks and become the mainstream. Thus this paper is focused on Transformer-based PLMs. The representative PLM infrastructures mainly include Transformer [Vaswani et al. 2017] or its variants such as Transformer-XL [Dai et al. 2019], Longformer [Beltagy et al. 2020], and

Table 1. A general overview of the tasks involving CTG. We enumerate 7 categories of tasks involving CTG, and briefly describe the input and output of each task, and list relevant representative works based on controllable aspects.

Task	Input & Output	Controllable Aspects
Attribute-based Generation	Input: Keywords, discrete attributes Output: Attributes-specific sentence	Topic [Dathathri et al. 2019; Khalifa et al. 2020; Tang et al. 2019; Wang et al. 2019a], tense [Hu et al. 2017b; Logeswaran et al. 2018],politeness [Sennrich et al. 2016], sentiment [Dathathri et al. 2019; Ghosh et al. 2017; Hu et al. 2017b; Khalifa et al. 2020] [Chen et al. 2019b; Logeswaran et al. 2018; Samanta et al. 2020; Zhang et al. 2019b], keywords [He 2021; Wang et al. 2021; Zhang et al. 2020d]
Dialogue Generation	Input: dialogue content, additional structural information(eg: persona, emotion, intent,etc.) Output: Dialogue Response	Persona [Li et al. 2016b; Mazaré et al. 2018; Siddique et al. 2017; Wolf et al. 2019] [Song et al. 2021, 2020b; Zhang et al. 2018b; Zhong et al. 2020], [Zeng and Nie 2021; Zheng et al. 2020] Politeness[Niu and Bansal 2018], Sentiment [Liu et al. 2019a; Lubis et al. 2018; Zhang et al. 2017; Zhou et al. 2018] [Firdaus et al. 2020; Ruan and Ling 2021; Song et al. 2019; Wei et al. 2019], Ground-truth [Dinan et al. 2018; Ghazvininejad et al. 2018; Qin et al. 2019; Wu et al. 2020]
Storytelling	Input: Story elements Output: Story paragraph	Story structure [Fan et al. 2018; Fang et al. 2021; Goldfarb-Tarrant et al. 2020], story ending [Luo et al. 2019; Peng et al. 2018; Tambwekar et al. 2018], topic [Feng et al. 2018; Wang et al. 2019b; Yang et al. 2019b] [Chang et al. 2021; Lin and Riedl 2021], persona [Liu et al. 2020b; Prabhumoye et al. 2019]
Data to Text	Input: Table/graph data Output: Natural language text data	Structural information [Puduppully et al. 2019; Ribeiro et al. 2020; Song et al. 2020a] [Ribeiro et al. 2021; Su et al. 2021; Zhao et al. 2020]
Data Augmentation	Input: Original text, predefined slot values Output: Text that specific features are replaced	Predefined slot values [Amin-Nejad et al. 2020; Liu et al. 2020c; Malandrakis et al. 2019]
Debiasing	Input: Biased text/biased model Output: Unbiased text/unbiased model	Predefined bias types(eg: political bias [Liu et al. 2021a] gender bias [Dinan et al. 2019; Qian et al. 2019], subjective bias [Pryzant et al. 2020], social bias [Barikeri et al. 2021; Sheng et al. 2020], sentiment bias [Huang et al. 2019]), toxicity [Krause et al. 2020; Liu et al. 2021c])
Format Control	Input: Desired formation, prompt text Output: Text in predefined formation	[Li et al. 2020b; Liao et al. 2019; Luo et al. 2016; Zhang and Lapata 2014] [Shao et al. 2021; Sheng et al. 2021]

Table 2. An overview of the characteristics of typical PLMs. "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 details of pre-trained task's definition could be seen in literatrue [Liu et al. 2021d]

Name	Model Type	Infrastructures	Pre-trained Task	Main Application
BERT [Devlin et al. 2018]	AE	Encoder	MLM+NSP	NLU
XLNET [Yang et al. 2019a]	AR	Transformer-XL	PLM	NLU
GPT2 [Radford et al. 2019],GPT3 [Brown et al. 2020]	AR	Decoder	SLM	NLG
T5 [Raffel et al. 2019]	Seq2Seq	Encoder+ Decoder	CTR	NLG+NLU
mBART [Liu et al. 2020a]	Seq2Seq	Encoder+ Decoder	FTR	NLG
UniLM [Dong et al. 2019]	AE+AR+Seq2Seq	Encoder+Decoder	SLM+CTR+NSP	NLG+NLU
ERNIE-T [Zhang et al. 2019a]	AE	Encoder	CTR+NSP	NLU
XLM[Lample and Conneau 2019]	Seq2Seq	Encoder+ Decoder	TLM	NLG
CPM[Zhang et al. 2020a]	AR	Decoder	SLM	NLG

Reformer [Kitaev et al. 2020], etc. The objectives of model learning mainly include masked language modeling (MLM) and next sentence prediction (NSP), etc. In order for a good understanding of them, we give a summary of representative PLMs in Table 2 according to their different data construction modes, model infrastructures, and pre-trained tasks, etc. 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 PLMs 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 do not require any masking 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 and sequence labeling, etc., which are more inclined to natural language understanding (NLU) tasks.

Auto-Regressive (AR) Models: The same as the classical language modeling approach, the main task of AR models is to predict the next word based on what has been read in text. A representative of this type of model is the GPT family. Unlike the aforementioned AE language models, the infrastructures of AR are composed of the Transformer's decoder part, and they use a masking mechanism in the training phase so that the attention calculations can only see the content before a word, but not the content after it. 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 seq2seq models use both encoder and decoder of the Transformer, for a better model flexibility. Currently, the most representative models of this type include T5[Raffel et al. 2019] and mBART[Liu et al. 2020a]. In principle almost all pre-trained tasks used in AE and AR models can be adapted to the seq2seq models. Relevant research [Raffel et al. 2019] has found that seq2seq models can achieve a 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 [Raffel et al. 2019], including sentence classification, semantic similarity matching, etc.

When it comes to the controlled text generation using PLMs, most of the methods exploit the generative model including AR and Seq2seq models as a basis, and then 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:

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

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

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

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 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 large-scale corpus that can help can produce more fluent and 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, making it difficult to interpret the generated text and lacking controllability. How to make a 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 on the main approaches in this area.

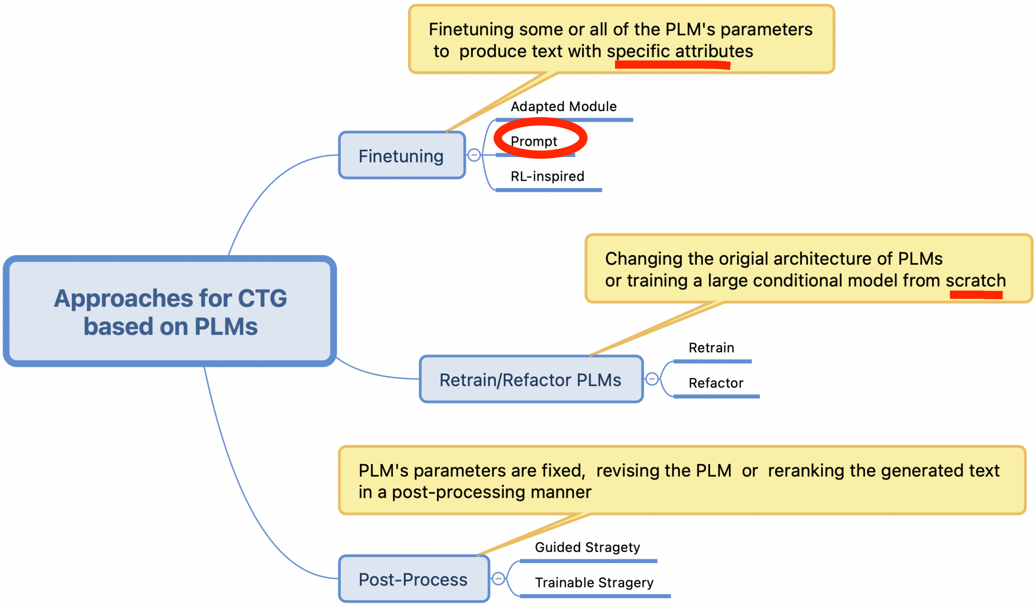

Finetuning some or all of the PLM's parameters to produce text with specific attributes
Adapted Module
Prompt
RL-inspired
Finetuning
Changing the origial architecture of PLMs or training a large conditional model from scratch
Approaches for CTG based on PLMs
Retrain
Refactor
Retrain/Refactor PLMs
PLM's parameters are fixed, revising the PLM or reranking the generated text in a post-processing manner
Guided Stragety
Trainable Stragery
Post-Process

Fig. 3. The Overview of the approaches of CTG based on PLM. According to the way how the control signal work with the pre-trained language model, we have roughly divided the existing methods into three categories, each of which is further divided into several subclasses.

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 the way 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 3. The most direct way is to **fine-tune** the PLMs, which can perform the controllable generation task at a lower cost. The second way is to **retrain or refactor** the PLMs for controlled text generation. This method could produce better results in principle, but may consume a lot of computing resource and also face the problem of lacking labelled data. As the parameter size of PLMs increase rapidly, even fine-tuning has become resource-intensive. To tackle the problems, text generation methods that work on the decoder, named as **post-processing**, have emerged, where PLMs are 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, an increasing attention from academic community has been paid to this direction in recent years. In the following sections, we will review the related literatures appeared in recent years and discuss these three types of methods in more detail.

3.2 Fine-tuning

This type of methods aim to fine-tune part or all of the PLMs' parameters to produce text that satisfies the specific constraints. As discussed in Section 2.3, "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 text into a PLM. Then a domain/task-adapted model will be obtained to achieve competitive performance by fine-tuning the PLM based on a small amount of labelled data for the specific downstream task.

This paradigm is also applicable to controllable text generation, and a large number of related studies have been carried out. Recent work has found that fine-tuning PLMs on target data such as AMR-to-text [Kale 2020; Radev et al. 2020; Ribeiro et al. 2020] for dialogue generation can establish a new level of performance. While the conventional fine-tuning method is relatively concise and easy to understand, we focus on more advanced methods below.

Adapted Module: This method constructs a task-related adapted network module around a PLM, and then it is trained with the PLM on the target dataset just like usual fine-tuning. Auxiliary Tuning [Zeldes et al. 2020] introduces an extra condition modeling module based on the original PLM, which takes $X(x_{<t};\alpha)$ as input and outputs logits in the vocabulary space at every token x_t. The auxiliary model is trained by adding its logits to the PLM's logits and maximizing the likelihood of the target task output. [Ribeiro et al. 2021] add an adapter module after the feed-forward sub-layer of each layer on both encoder and decoder of the PLM, which can encode the graph structure into the PLMs without contaminating its original distributional knowledge. During the training stage, the PLM's parameters are frozen, and only the injected adapter is trainable. Avoiding catastrophic forgetting 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 adapted module is also applied. Lin et al. propose an adapter-bot for dialogue generation. The model builds a series of lightweighted adapters on top of a PLM for dialogue generation, namely, DialGPT [Zhang et al. 2020c]. 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 rational way for the use PLMs is to keep the fine-tuning phase's training objective consistent with the original task where the PLMs are derived. 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". Different from 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 sentence's sentiment is [MASK]**"), and then ask the model to predict the token [mask] according to the original training task for the PLM. This approach has gone through various stages, from manual template construction [Jiang et al. 2020], to automated search for discrete tokens [Shin et al. 2020], to continuous virtual Tokon representations [Lester et al. 2021; Li and Liang 2021]. It has achieved a great success in few-shot scenarios.

From the CTG point of view, the prompt-based approach still applies. Li and Liang [Li and Liang 2021] propose a method named “prefix tuning”, which freezes the PLM's parameters and back-propagates the error to optimize a small continuous task-specific vector called "prefix". The learnt 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, namely P-tuning [Lester et al. 2021], serves a similar purpose. Different from prefix-tuning [Li and Liang 2021], p-tuning does not place prompt with the "prefix" in the input, but constructs a suitable template to prompt the PLM, and the template is composed of continuous virtual token which is obtained through gradient descent.

More recently, Zou et al. [2021] propose a method called Inverse Prompt, the main idea of which is to use generated text candidates of 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 provide 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.

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

Reinforcement Learning (RL) inspired Approaches: The core motivation of this type of methods 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. [2019] 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 $\rho(y_i|x)$. Human labellers 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 ρ 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:

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

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

Liu et al. [2020c] propose a data augmentation approach, which uses reinforcement learning to guide the GPT2 model to generate texts towards a specified conditional direction (i.e, target class). Specifically, an additional RL stage is added between the softmax and argmax functions of GPT2, and 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. [2020] use a RL-based approach on the task of English summarization, which fine-tunes the PLM by combining with human feedbacks.

The reinforcement learning is also applied for controllable story generation. Tambwekar et al. [2018] designs 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 LSTM, but its principles are applicable to the subject described in this article, so that we have included it here.

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

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 the insufficient labeled data and large computing resource consumption.

CTRL [Keskar et al. 2019] is an early attempt in this direction. It trains a language model conditioned on a variety of control code. The network model used in this approach is also 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 lets the original language model $p\left(x_i \mid x_{<i}\right)$ transform into $p\left(x_i \mid x_{<i}, c\right)$. A language model with 1.63 billion parameters is retrained on a 140Gb 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. [2020d] propose POINTER, an insertion-based method for hard-constrained (keep the specific words appeared in generated text) text generation. Different from the auto-regressive method such as GPT2, this method modifies the structure of 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 condition, 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 methods can ensure the generated sentences to 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 idea of insertion-based method, like POINTER [Zhang et al. 2020d], a lexically constrained text generation framework called Constrained BART (CBART) is proposed [He 2021]. This approach also adopts the way of progressive insertion/replacement for text generation, yet without the modification of 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 so as 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) [Chan et al. 2020] introduces a conditional control module in addition to the original pre-trained language model, which can realize the precise control of the generated text at the word- and phrase- levels. In term 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 labelled 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 in a piece of text as 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 content inputs 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. 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, and it is formally represented as mention status embeddings injected into the Transformer's decoder, 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. propose a pre-training based method to build a personalized dialogue agent. The whole framework is in a encoder-encoder (Transformer) fashion, and its initial parameters inherit from an existing PLM model. The personalized information is represented as attribute embeddings, which are 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 the historical dialogue information dynamically. To solve the problem of the lack of labeled data in conditional dialogue generation, a multi-task learning framework is proposed [Zeng and Nie 2021], which utilizes both conditional labeled dialogue data and non-dialogue text data. Based on a condition-aware transformer block (reconstructed from the original Transformer), three subtasks 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 has achieved the state-of-the-art performance so far.

In summary, this type of approaches refactor or retrain the PLMs to achieve controlled text generation. They are more flexible and convenient to use, but may make the PLM lose its 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. In the realm of controlled text generation, the idea of fixing the PLM's parameters firstly and reranking the generated text in a post-processing manner becomes achievable and promising.

The most natural idea about the post-processing method is to use some common decoding algorithms in text generation, e.g., the Greedy search, constraint beam search [Anderson et al. 2017], Top-k sampling [Fan et al. 2018], Nucleus sample [Holtzman et al. 2019b], 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 methods decouple 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 [Dathathri et al. 2019]. It firstly trains an attribute discriminant model and then uses it to guide 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 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 the structure or retrain the PLM, and it is able to achieve a significant improvement in attribute alignment. However it has a slight deficiency in text fluency measured with the metric of PPL (Perplexity).

MEGATRON-CNTR [Xu et al. 2020] 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 (GPT2) with the story context to get the next sentence. The entire process is repeated. Human evaluation results show that up to 91.5% of the generated stories are successfully controlled by the new keywords. In this framework, GPT2 is independent of the other modules and generates context-relevant sentence using introductory text provided by MEGATRON-CNTR's component as input, without any adaptation in training.

Hua and Wang [2020] propose a content-controlled text generation framework, namely FAIR. It uses the BERT [Devlin et al. 2018] model to automatically construct a content plan, including keyword assignments and their corresponding sentence-level positions. After that, the BART [Liu et al. 2020a] without structure modification is applied 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 keyphrases and the generated texts.

Additionally, a series of discriminator-guided ways have been developed, which train an attribute discriminator to help PLM select text for the specific attributes when decoding. Adversarial Search [Scialom et al. 2020], inspired by GAN (Generative Adversarial Network), trains the discriminator to distinguish human created text from 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 thousands words, decoding using a discriminator to classify each token is time-consuming. Aimed at solving this problem, GeDi [Krause et al. 2020] trains a small class-conditional language model (CC-LM) as generative discriminators to guide the generation from large PLM (GPT2, GPT3). Specifically, an 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, 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 $\bar{c}$, and 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 an original PLM. A contrast mechanism is designed to compute the probability that every candidate next token x_t belongs to the desired class, given by $P_\theta(C \mid x_t, x < t)$:

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

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

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

Since the calculation of the Equation 8 only needs two parallel forward passes of CC-LM, the generation efficiency is greatly improved.

Inspired by the GEDI [Krause et al. 2020], a batch of similar work have emerged. DEXPERTS [Liu et al. 2021c] re-ranks the predictions of the PLM based on expert (and anti-expert) opinions during the decoding stage to steer the language model towards the desired generation. FUDGE [Yang and Klein 2021] learns an attribute predictor operating on a partial sequence to adjust the original PLM's probabilities, and obtain 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 [Lin and Riedl 2021] extends the GEDI model to controlled story generation by introducing a planner module.

More recently, Pascual et al. [2021] propose a simple yet efficient plug-and-play decoding method, namely K2T, which even does not need a discriminator. Specifically, given a topic or keyword which is considered as hard constraints, K2T add a shift to the probability distribution over the vocabulary towards the words which are semantically similar to the target constraint word. The shift is calculated based on the 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.

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.

Trainable Strategies: Different from Guided Strategies, although the methods using the trainable strategy also work in the inference phase, the extra processing module needs to be trained jointly with PLM whose parameters are fixed.

Energy Based Model (EBM) [Deng et al. 2020] is proposed to guide the PLM to generate desired text. The generative model is formalised 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 to steer the joint model $P_\theta(x)$ to get close to 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 yield 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 [Khalifa et al. 2020] was proposed to solve the problem. It uses the Energy based Model (EBM) to represent the point-wise and distributional constrains in one go:

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

Table 3. A summary of each CTG method. We list the main characteristics of three categories of CTG methods from a macro perspective, and list the typical literatures in seven sub-categories.

Method	Main characteristics	Subcategory	Typical References

Fine-tuning	standard training;
	efficient inference;
	higher text quality;
	weaker controllability
		Adapted Module	[Zeldes et al. 2020], [Ribeiro et al. 2021],[Lin et al. 2021]

		Prompt	[Jiang et al. 2020],[Shin et al. 2020],[Lester et al. 2021], [Li and Liang 2021]
		Reinforce Learning	[Ziegler et al. 2019],[Stiennon et al. 2020],[Liu et al. 2020c], [Tambwekar et al. 2018]
Refact/Retrain	computationally expensive training; higher text quality; better controllability	Retrain	[Keskar et al. 2019],[Zhang et al. 2020d]
		Refact	[Zhang et al. 2020d],[Chan et al. 2020],[Wang et al. 2021],[He 2021] [Zheng et al. 2020],[Zeng and Nie 2021]

Post-Process	efficient training;
	computationally expensive inference;
	lower text quality; better controllability
		Guided Strategy	[Fan et al. 2018],[Holtzman et al. 2019b], [Dathathri et al. 2019],
			[Xu et al. 2020], [Scialom et al. 2020],[Krause et al. 2020],
			[Liu et al. 2021c],[Yang and Klein 2021],[Lin and Riedl 2021]
			[Pascual et al. 2021]

		Trainbale Strategy	[Deng et al. 2020],[Khalifa et al. 2020]

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

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

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, the approach adopts a method similar to variational inference to approximate the distribution, by first initializing a policy $\pi = a(x)$, and then minimizing the cross entropy between the policy π and $p(x)$:

$$CE\,(p, \pi_\theta) = -\sum_{x} p(x) \log \pi_(x). \tag{14}$$

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 shows a superiority in satisfying the constraints while avoiding degeneration. However, this approach suffers from a high computational cost. This challenge needs to be addressed in the future.

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 methods build upon the probability modeling theory and thus has a good theoretical basis, yet they are still at the early stage and need to improve the issues related to computational efficiency and text quality.

3.5 Summary

In this section, we divide the current PLM-based CTG approaches into 3 categories according to the way how PLMs are used. For each category, we analyze its main principles, process and methods. A summary of the works is shown in Table 3 . "Fine-tuning" is a more general 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 cost and the lack of large scale labelled data. To overcome the limitations, combining general pre-trained models and the use of semi-supervised or self-supervised learning to build a pre-trained model dedicated to CTG, would be a feasible way in the future.

The emergence of post-processing methods is rooted on the powerful text generation capabilities of pre-trained language models. This kind of methods generally assume that the PLMs can produce high-quality text, and then use a post-processing module as a filter to screen out the desired type of text. Most current decoding-time approaches (post-process) are still computationally expensive 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 model training stage. In recent years, the scale of pre-trained models is getting larger, and their mastery of the 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.

4 EVALUATION METHODS

After building an NLG model, we need to assess its performance, for which variou evaluaion methods can be used. The performance of a model is reflected by suitable evaluation metrics. CTG is different from the general NLG tasks due to the need of fulfilling 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, when evaluating a CTG model, we usually use both general and CTG-specific metrics. And we will discuss these metrics in detail below.

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 [Celikyilmaz et al. 2018; Du et al. 2017], 2) **factuality**: to what extent the generated text reflects the facts described in the context. [Holtzman et al. 2019a; Welleck et al. 2019], 3) **grammar**: how grammatically correct the generated text is, 4) **diversity**: whether the generated text is of different types or styles. The ways of measuring these general evaluation aspects can be divided into three categories based on who makes the judgments on them: human beings or the machine (as shown in Figure 4).

4.1.1 Human-Centric Evaluation Metrics. Natural Language is created by human beings as a crucial form of human communication. So humans are the best evaluators of the natural language texts generated by NLG systems. We call the evaluation metrics that only human beings involve as human-centric evaluation metrics and they 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 5(a), RankME in Figure 5(b), etc. [Celikyilmaz et al. 2018; Holtzman et al. 2019a; Novikova et al. 2018].

Indirect evaluation. Different from 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 [Gkatzia and Mahamood 2015] which are usually measured by the User Task Success evaluation), or from a system's perspective [Aziz et al. 2012; Denkowski et al. 2014], such as the performance of a dialogue system which is usually measured by the System Purpose Success evaluation.

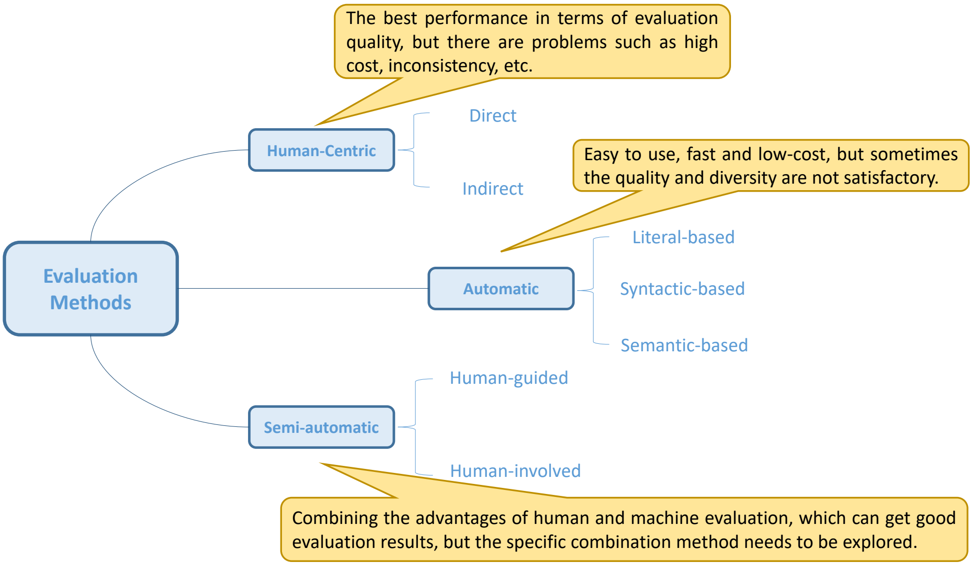

Evaluation Methods
Human-Centric
Direct
Indirect
The best performance in terms of evaluation quality, but there are problems such as high cost, inconsistency, etc.
Automatic
Literal-based
Syntactic-based
Semantic-based
Easy to use, fast and low-cost, but sometimes the quality and diversity are not satisfactory.
Semi-automatic
Human-guided
Human-involved
Combining the advantages of human and machine evaluation, which can get good evaluation results, but the specific combination method needs to be explored.

Fig. 4. An overview of general evaluation metrics divided by the participants.

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 benchmarking datasets. We divide the similarity measures into three categories: lexical-based, syntactic-based, and semantic-based. For the convenience of explanation, we use a pair of sentences **"the man was found on the chair"** (a generated text) and **"the man was on the chair"** (the corresponding reference text), as an example to illustrate the evaluation metrics in the rest of this section.

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 the overall sentence-level similarity. Based on the different granularity levels of the basic lexical unit, the metrics can be further categorized:

1) n-gram [1] level:

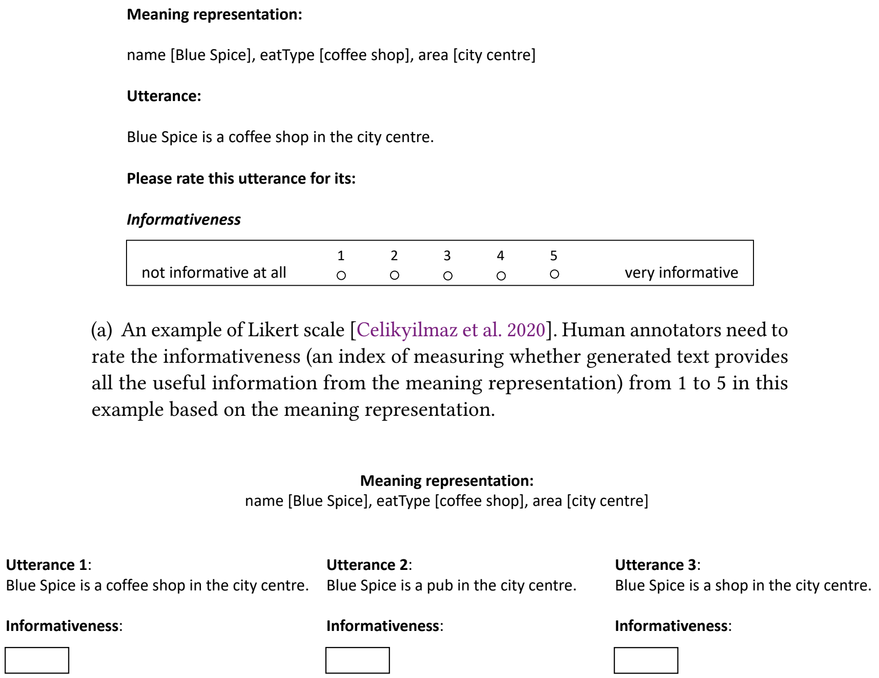
BLEU [Papineni et al. 2002] is a commonly used metric in natural language processing tasks to evaluate the difference between generated texts of an NLG model and reference texts. Its value fall into the range between 0.0 and 1.0. If two sentences match perfectly, the value of BLEU is 1.0. If there is no overlap between them at all, the BLEU value is 0.0.

Specifically, BLEU counts the n-gram matches in the generated text with the reference text. For convenience, we use BLEU-n to represent BLEU with respect to n-grams, as follows.

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

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

[1] n-gram is a statistical language model algorithm that groups every n tokens in the text to form a sequence of fragments with the length of n.



Meaning representation:

name [Blue Spice], eatType [coffee shop], area [city centre]

Utterance:

Blue Spice is a coffee shop in the city centre.

Please rate this utterance for its:

Informativeness

	1	2	3	4	5	
not informative at all	○	○	○	○	○	very informative

(a) An example of Likert scale [Celikyilmaz et al. 2020]. Human annotators need to rate the informativeness (an index of measuring whether generated text provides all the useful information from the meaning representation) from 1 to 5 in this example based on the meaning representation.

Meaning representation:
name [Blue Spice], eatType [coffee shop], area [city centre]

Utterance 1:
Blue Spice is a coffee shop in the city centre.

Informativeness:

□

Utterance 2:
Blue Spice is a pub in the city centre.

Informativeness:

□

Utterance 3:
Blue Spice is a shop in the city centre.

Informativeness:

□

(b) An example of RankME [Celikyilmaz et al. 2020]. Human annotators need to give a score of the informativeness in a given range for each utterance in this example based on the meaning representation.

Fig. 5. Examples of fine-grained direct evaluations of human-centric evaluation metrics Likert scale and RankME.

Take BLEU-2 as an example. the set of 2-grams in the above example generated text, i.e., G, is {the man, man was, was found, found on, on the, the chair}, and in the reference text, i.e., R, is {the man, man was, was on, on the, the chair}. According to Equation 15, BLEU-2 in this example is $4/6 = 2/3$.

The n-grams in BLEU is simple to use and fast to calculate, but it ignores the structure and semantic information of the text. Furthermore, as an extension for BLEU, **Self-BLEU** [Zhu et al. 2018] 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 generated text, and takes the average BLEU score to represent the diversity of all generated texts. The generated text with a higher diversity has a lower self-BLEU score.

ROUGE [Lin 2004] is the abbreviation of *Recall-Oriented Understanding for Gisting Evaluation.* It is an automatic summary evaluation method, which is a set of indicators to evaluate the quality of the generated texts automatically. It compares the generated texts with a group of reference texts, counts the number of overlapping basic units (n-gram), 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:

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

Comparing the 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. While 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. For the example above, Equation 16 gives the ROUGE-2 of 4/5.

Perplexity (PPL) is a word-level method to measure the advantages and disadvantages of a language probability model. A Language probability model is a probability distribution on 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, when a language probability model is trained on the training set with 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:

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

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 the i-th word in a language model trained with reference texts. The smaller PPL value, the better fluency of the generated text.

In addition, a variant of PPL called **Reverse PPL** [Shi et al. 2020] mainly focuses on the diversity of 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 [Li et al. 2016a] is an n-gram based metric and applies to some scenes that pursue the diversity of generated texts (such as dialogue, advertisement generation). The greater the value, the higher the diversity. It is formulated as follows:

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

where the numerator is the number of n-grams that appear once in the generated texts and the denominator is the total number of n-grams in the generated texts. In the above example, since there is no duplicated n-gram in the generated text, i.e., the count of unique n-grams equals the total number of n-grams, the value of Distinct-2 is 1.

2) word token level:

Word Mover's Distance (WMD) [Kusner et al. 2015] measures the semantic distance between a pair of texts. It consists of three steps: 1) express all words w in both generated and reference texts as word2vec $\mathbf{w}$ [Mikolov et al. 2013b]; 2) for each word $\mathrm{w_g}$ in the generated text, we first find a corresponding word $\mathrm{w_r}$ in the reference text; 3) calculate the moving distance between $\mathrm{w_g}$ and $\mathrm{w_r}$ by the European distance and find the smallest sum of the moving distances from all words of the generated text to all words of the reference text. The definition of this distance is consistent with the famous transportation problem *Earch Mover's Distance* [Rubner et al. 2000]. The formula of WMD is as follows:

$$min_{T \geq 0} \textstyle\sum_{i,j=1}^{n} E(w_i, w_j), \tag{19}$$

where $E(w_i, w_j)$ is the Euclidean Distance between word w_i and word w_j, T is a transfer matrix of words in sentences that need to be compared.

MEANT 2.0 [Lo 2017] is a new version of MEANT [Lo and Wu 2011]: a vector-based semi-automatic similarity measure that uses word embedding and shallow semantic parsing to calculate lexical and structural similarity instead of using human annotators. Furthermore, MEANT 2.0

adopts idf[2]-weighted distributional n-gram accuracy to measure the phrasal similarity of semantic framework and its role filler between the reference and the generated texts.

3) sentence level:

Editing distance [Yujian and Bo 2007] is a quantitative measurement of the difference between two strings (they mean the reference and the generated texts in NLG tasks). The principle of the method is to measure how many times it takes to change one string into another:

$$lev_{r,g}(i,j) = \begin{cases} max(i,j) & if \ \min(i,j) = 0, \\ min \begin{cases} lev_{r,g}(i-1,j)+1 \\ lev_{r,g}(i,j-1)+1 \\ lev_{r,g}(i-1,j-1)+1_{(r_i \neq g_j)} \end{cases} & otherwise, \end{cases} \tag{20}$$

where i, j are the i-th/j-th letter in the reference and the generated text, respectively. $lev(\cdot)$ is the Levenshtein Distance function with the following editing operations: insert, delete and replace. In the example mentioned above, the editing distance between the generated and the reference sentences is 5, as we need to delete five letters (i.e., the word "found") in the generated text in order to change it to the reference text.

Syntactic-based Metrics. Syntax is related to the grammatical arrangement of words in a sentence. Common syntactic-based metrics are as follows:

TESLA [Liu et al. 2010] is the abbreviation for *Translation Evaluation of Sentences with Linear-programming-based Analysis*. The goal of it is to output a score to measure the quality of the generated texts. TESLA-M [Dahlmeier et al. 2011], a lighter version of TESLA, is the arithmetic average of F-measures between bags of n-grams (BNGs). It first matches BNGs with two different similarity functions: maxSim and POS (part-of-speech) tag. Then, we combine different n-grams and similarity functions and get a score based on them to measure the quality of the generated texts. TESLA is an advanced version that adopts a general linear combination of three types of features including TESLA-M. There are also various other syntactic-based metrics, for example Liu and Gildea [2005] adopts the constituent labels and head-modifier dependencies in the evaluation.

Semantic-based Metrics. Semantic-based metrics aims to handle the evaluation of texts that are lexically different but have a similar semantic meaning. Compared with lexical-based and syntactic similarity, semantic similarity requires more information to be considered and is more difficult to measure.

Typical semantic similarity measures include: **Semantic Textual Similarity (STS)** [Agirre et al. 2016] which adopts weighted Pearson correlation to measure whether the underlying semantics of the generated texts are equivalent to the corresponding human-written texts; **Deep Semantic Similarity Model (dssm)** [Huang et al. 2013] which measures the similarity by computing the distance between vector representations of texts that are projected onto a low-dimensional space by latent semantic models.

Naturally, machine learning-based methods can also applied to learn metrics between a pair of sentences. Some early works are mainly focused on calculating the matching scores between texts by training an end-to-end neutral model or regression model [Chen et al. 2017; Shimanaka et al. 2018; Stanojević and Sima'an 2014]. In recent years, PLM-based evaluation methods have emerged. BERTscores [Zhang et al. 2020b] replaces the n-gram overlaps defined in BLEU, with embeddings from BERT, to learn a semantic-awareness metric. Similarly, Bertr and YiSi [Mathur et al. 2019] take the advantage of BERT embeddings to capture the semantic similarity between the generated text and its reference. Some approaches also try to fine-tune PLMs for text quality

[2] idf is short for inverse document frequency

estimation [Liu et al. 2021b; Reimers et al. 2019; Zhou and Xu 2020]. Particularly, Sellam et al. [2020] propose a task-specific pre-trained model, namely 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.

To measure the performance of an NLG model more generally and comprehensively, a number of evaluation benchmarks have been proposed including multiple tasks. [Wang et al. 2018] proposed the General Language Understanding Evaluation (GLUE) benchmark which contains nine different tasks to evaluate the model's understanding and mastery of general language information. Similarly, [Liu et al. 2020d] proposed the General Language Generation Evaluation (GLGE), a new multi-task benchmark for natural language generation. DecaNLP [McCann et al. 2018] and GEM [Gehrmann et al. 2021] are also well-known benchmarks for NLG tasks.

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 less ideal, because they can mainly evaluate the surface and simplex similarity between the reference and target sentences. Humans can better distinguish a more diverse range of features such as fluency and grammar, so that semi-automatic evaluation combines automatic and human-centric evaluation methods to make a better use of the benchmarking datasets for these harder tasks, in order to get more reliable evaluation results. Semi-automatic evaluation metrics can be divided into two categories according to the different ways of using human evaluation results:

Human-guided evaluation. This type of method directly treats the results of human evaluation as the training target (i.e., labels) of the model. Lowe et al. [2017] encode the context, the generated text and a reference text into vectors (**c**, **g**, and **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 a good consistency with human judgments, but sometimes it may be too conservative and lacks diversity due to the quality of human evaluators.

Human-involved. Without using a human judged score as label, we can simply calculate the perplexity of a probabilistic model first, and then let humans evaluate the beam-searched outputs. More complicatedly, Hashimoto et al. [2019] 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 extra evaluation metrics that take into account the controlled elements. Adherence measurement can be used for this purpose, and there are some typical ways to measure the effectiveness of a CTG model.

Accuracy. With this type of method, 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. Accuracy, which is equal to the number of test samples in the source domain divided by the number of test samples correctly accord with controlled elements, can measure the performance of the model [Chen et al. 2019a; Fu et al. 2018; Gao et al. 2019]. It can be applied to a large-scale dataset but will become less reliable when facing unbalanced data.

Keyword-based. Measures such as point-wise mutual information (PMI) are used to measure the correlation between the controlled reference texts and generated texts [Syed et al. 2020; Wang et al. 2019a]. The PMI is calculated as:

$$PMI(x, y) = log_2 \frac{p(x, y)}{p(x)p(y)}. \tag{21}$$

The greater the correlation between the word x in controlled reference text and the word y in generated text, the greater the PMI.

The word-level KL divergence (wKL) can also measure the quality of generated sentences by calculating the KL divergence between the distribution t of word x frequency in controlled training data and the distribution g of word y frequency in generated data [Shi et al. 2020], which formulated as follows:

$$KL(t||g) = \sum_i t(x) \frac{t(x)}{g(y)}. \tag{22}$$

LM-based. Language models trained on each controlled training data are adopted to score the generated sentences by negative log-likelihood values. Ideally, for the sentence generated under a certain controlled element, the corresponding language model will output a score that is different from the others and has the lowest negative log-likelihood value. Therefore we can judge whether it conforms to a certain controlled element [Tikhonov and Yamshchikov 2018; Vechtomova et al. 2018]. This approach is not suitable for tasks that lack controlled training data for training of the language models.

Jensen-Shannon Divergence (JS Divergence). JSD is a symmetrical divergence based on Kullback-Leibler divergence (KL Divergence). Its formulation is as follows:

$$JS(t||g) = \frac{1}{2} KL(t|| \frac{t+g}{2}) + \frac{1}{2} KL(t|| \frac{t+g}{2}). \tag{23}$$

This metric can measure the difference between controlled training data probability distribution t and generated data probability distribution g. Furthermore, because the value range of JSD is from 0 to 1, we can adopt it in interpolation evaluations [Ghabussi et al. 2019]. Ideally, when generating a text controlled equally on two elements, we can get 0.5 for these two elements and 0 for other elements.

4.3 Summary

The evaluation metrics discussed above need to be flexibly combined according to different tasks. They have their advantages and disadvantages. Sometimes, multiple evaluation methods can be used together to determine the performance of an NLG model.

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 of evaluators, selection of personnel, cost of the evaluation and other steps all require time and manpower, especially the evaluation tasks that can only be handled by domain experts; 2) **Maintaining quality control** [Ipeirotis et al. 2010; Mitra et al. 2015]: although the emergence of online crowd-sourcing platforms has eased the cost problem to a certain extent, we cannot guarantee the quality of personnel conducting online evaluations;

3) **Lack of consistency** [van der Lee et al. 2020]: due to the change of evaluation personnel, the reproducibility of the evaluation results may not be high, which leads to the inconsistency problem.

Compared with the human-centric evaluation metrics, automatic evaluation metrics are easy to use, fast to obtain results, and are low-cost. However, the evaluation of this type is less precise than human assessments. It is vital to develop automated evaluation methods comparable to the human level in the future.

Semi-automatic evaluation metric combines the advantages of human-centric methods and automatic methods, but still require a lot of human judgments or labeling, which is also 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 above general evaluation metrics to evaluate the generated texts 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, we think that the challenges of text quality evaluation still lie in the general evaluation methods.

5 CHALLENGES AND FUTURE DIRECTIONS

5.1 Challenges

The birth of large-scale pre-trained language models brings the research of text generation to a new stage. The PLMs have mastered a remarkable level of linguistic knowledge (semantic, syntax, etc.) from large-scale corpus, enabling the production of more fluent and diverse text naturally. 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 make full use of the powerful PLMs to generate the desired and controllable text, has become a promising yet challenging new research field in both academia and industry. Based on the systematic review of the key concepts, methods and findings in the latest development of PLM-based controllable text generation, we think this promising and fast growing area is still facing a number of challenges in the following aspects.

First, pre-trained language models have learned rich knowledge from large-scale corpus used for pre-training. However, a 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 make the PLM-based NLG model 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 in 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, and there is also a lack of information interaction between them, leading to a coarser granularity in the guidance process and decreased quality of the generated text.

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 have certain limitations in paragraph/document-level modeling. For example, it is hard to keep long-range coherency in both semantic logic and controlled condition for long text generation. It calls for further research to establish 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 pre-trained language models are typically data-driven, which allows the models to learn the primary logic and common sense contained in the training corpus. However, the knowledge captured in those models is rather superficial. The PLMs will lose generalization ability when the training data does not contain relevant common sense and domain-specific knowledge. Therefore, purely relying on PLMs could be difficult to control the generated texts faithfully to the common sense and rich knowledge specific to the target domain.

Fifth, a 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 generated text of a NLG model cannot be accurately evaluated, it is hard to think about a way to control them.

Finally, we believe that the research on controlled text generation is still in an early stage. In Section 2.2 of this paper, 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 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 on 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. At present, most of the Prompt-based methods are applied to NLU tasks, but there is relatively little work on NLG tasks. Therefore, we believe there is an urging need to explore prompt-based learning methods specifically for CTG.

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 a more effective control. For example, allowing more information interaction between the guided model and the generative model will ensure finer-grained text generation and higher text quality. 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 the decoding phrase, achieving the simultaneous control of multiple aspects for a generated sentence.

Integration with Classic Generative Theory and Linguistic Knowledge: The controllable text generation task can be regarded as obtaining, from 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) [Goodfellow et al. 2020], Variational Autoencoders (VAEs) [Kingma and Welling 2013], Energy-based Model [Deng et al. 2020] and their variants. It is known that auto-regressive PLMs, i.e., the GPT family models, can not model global information of the generated texts naturally, making it difficult to control the generated text's distribution at the paragraph/doc level. We expect that combining with 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 fashion, 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/chapter structures and logic are needed in long text generation, which PLMs can not 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, so as to better ensure the quality and controllability.

Incorporation of Knowledge Graphs: Knowledge graph is a natural carrier of explicit common sense and domain-specific knowledge, and 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. A straightforward way is to convert the knowledge graphs into texts first, and then treat them as training corpus for the pre-training process of PLMs, so that the PLM models would have learnt the corresponding domain knowledge. In addition, knowledge graphs can also be introduced in the downstream tasks. We can regard knowledge graphs as constraint conditions, and filter out the texts that do not meet the constraint conditions during the decoding phrase of PLMs, ensuring the controllability of the generated text in term of common sense and logic.

Novel Evaluation Metrics and Methods: Developing innovative evaluation metrics for text generation 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 a NLG model would be an interesting and fascinating area for further investigation.

New CTG tasks: Controllable text generation is a relatively broad concept. In addition to the existing common controlled element of generated text, such as content, attributes and format, the connotation and the form of control elements can be re-defined and new tasks can be proposed according to new application scenarios in the future. For instance, we could construct corresponding syntax templates to make the text's syntax manageable, or combine with knowledge graphs including logical rules to make the generated text better conform to the prescribed logic, so as to achieve a good logical control. Furthermore, single-attribute control could be extended to any other aspects, such as humor text generation.

6 CONCLUSIONS

In this paper, 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 it. We are hopeful that this literature survey is able to provide a clear picture of the field and set a roadmap for researchers to move forward.

REFERENCES

Eneko Agirre, Aitor Gonzalez Agirre, Inigo Lopez-Gazpio, Montserrat Maritxalar, German Rigau Claramunt, and Larraitz Uria. 2016. Semeval-2016 task 2: Interpretable semantic textual similarity. In *SemEval-2016 Task 2: Interpretable semantic textual similarity. SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 512-24.* ACL (Association for Computational Linguistics).

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*. 4699–4708.

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

Wilker Aziz, Sheila Castilho, and Lucia Specia. 2012. PET: a Tool for Post-editing and Assessing Machine Translation.. In *LREC*. Citeseer, 3982–3987.

Soumya Barikeri, Anne Lauscher, Ivan Vulić, and Goran Glavaš. 2021. RedditBias: A Real-World Resource for Bias Evaluation and Debiasing of Conversational Language Models. *arXiv preprint arXiv:2106.03521* (2021).

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

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* (Virtual Event, Canada) *(FAccT '21)*. Association for Computing Machinery, New York, NY, USA, 610–623. https://doi.org/10.1145/3442188.3445922

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165 (2020). arXiv:2005.14165 https://arxiv.org/abs/2005.14165

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)*. 1662–1675.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of Text Generation: A Survey. *arXiv e-prints* (2020), arXiv–2006.

Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2020. CoCon: A Self-Supervised Approach for Controlled Text Generation. *CoRR* abs/2006.03535 (2020). arXiv:2006.03535 https://arxiv.org/abs/2006.03535

Haw-Shiuan Chang, Jiaming Yuan, Mohit Iyyer, and Andrew McCallum. 2021. Changing the Mind of Transformers for Topically-Controllable Language Generation. *arXiv preprint arXiv:2103.15335* (2021).

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

Huimin Chen, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, and Zhipeng Guo. 2019b. Sentiment-Controllable Chinese Poetry Generation.. In *IJCAI*. 4925–4931.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. ACL, Vancouver.

Daniel Dahlmeier, Chang Liu, and Hwee Tou Ng. 2011. Tesla at wmt 2011: Translation evaluation and tunable metric. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. 78–84.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* (2019).

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164* (2019).

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

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*. 395–404.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2019. Queens are powerful too: Mitigating gender bias in dialogue generation. *arXiv preprint arXiv:1911.03842* (2019).

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241* (2018).

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. *CoRR* abs/1905.03197 (2019). arXiv:1905.03197 http://arxiv.org/abs/1905.03197

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)*. 1342–1352.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical Neural Story Generation. *CoRR* abs/1805.04833 (2018). arXiv:1805.04833 http://arxiv.org/abs/1805.04833

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).

Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018. Topic-to-Essay Generation with Neural Networks.. In *IJCAI*. 4078–4084.

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. *arXiv preprint arXiv:1908.09022* (2019).

Jessica Ficler and Yoav Goldberg. 2017. Controlling Linguistic Style Aspects in Neural Language Generation. *CoRR* abs/1707.02633 (2017). arXiv:1707.02633 http://arxiv.org/abs/1707.02633

Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. 2020. EmoSen: Generating sentiment and emotion controlled responses in a multimodal dialogue system. *IEEE Transactions on Affective Computing* (2020).

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

Xiang Gao, Yizhe Zhang, Sungjin Lee, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2019. Structuring latent spaces for stylized response generation. *arXiv preprint arXiv:1909.05361* (2019).

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. *arXiv preprint arXiv:2102.01672* (2021).

Amirpasha Ghabussi, Lili Mou, and Olga Vechtomova. 2019. Stylized Text Generation Using Wasserstein Autoencoders with a Mixture of Gaussian Prior. *arXiv preprint arXiv:1911.03828* (2019).

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2017. Affect-lm: A neural language model for customizable affective text generation. *arXiv preprint arXiv:1704.06851* (2017).

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)*. 57–60.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. *arXiv preprint arXiv:2009.09870* (2020).

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.

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.

Tatsunori 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)*. 1689–1701.

Xingwei He. 2021. Parallel Refinements for Lexically Constrained Text Generation with BART. *CoRR* abs/2109.12487 (2021). arXiv:2109.12487 https://arxiv.org/abs/2109.12487

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019a. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019b. The Curious Case of Neural Text Degeneration. *CoRR* abs/1904.09751 (2019). arXiv:1904.09751 http://arxiv.org/abs/1904.09751

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

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017a. Controllable Text Generation. *CoRR* abs/1703.00955 (2017). arXiv:1703.00955 http://arxiv.org/abs/1703.00955

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017b. Toward controlled generation of text. In *International Conference on Machine Learning*. PMLR, 1587–1596.

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

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.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2019. Reducing sentiment bias in language models via counterfactual evaluation. *arXiv preprint arXiv:1911.03064* (2019).

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*. 64–67.

Zhengbao Jiang, Frank F. Xu, J. Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics* 8 (2020), 423–438.

Mihir Kale. 2020. Text-to-Text Pre-Training for Data-to-Text Tasks. *CoRR* abs/2005.10433 (2020). arXiv:2005.10433 https://arxiv.org/abs/2005.10433

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

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2020. A distributional approach to controlled text generation. *arXiv preprint arXiv:2012.11635* (2020).

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. *CoRR* abs/2001.04451 (2020). arXiv:2001.04451 https://arxiv.org/abs/2001.04451

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq R. Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. GeDi: Generative Discriminator Guided Sequence Generation. *CoRR* abs/2009.06367 (2020). arXiv:2009.06367 https://arxiv.org/abs/2009.06367

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*. PMLR, 957–966.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. *CoRR* abs/1901.07291 (2019). arXiv:1901.07291 http://arxiv.org/abs/1901.07291

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. *CoRR* abs/2104.08691 (2021). arXiv:2104.08691 https://arxiv.org/abs/2104.08691

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016a. 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*. 110–119.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155* (2016).

Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020b. 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

Shifeng Li, Shi Feng, Daling Wang, Kaisong Song, Yifei Zhang, and Weichao Wang. 2020a. EmoElicitor: An Open Domain Response Generation Model with User Emotional Reaction Awareness.. In *IJCAI*. 3637–3643.

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *CoRR* abs/2101.00190 (2021). arXiv:2101.00190 https://arxiv.org/abs/2101.00190

Yi Liao, Yasheng Wang, Qun Liu, and Xin Jiang. 2019. Gpt-based generation for classical chinese poetry. *arXiv preprint arXiv:1907.00151* (2019).

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

Zhaojiang Lin, Andrea Madotto, Yejin Bang, and Pascale Fung. 2021. The Adapter-Bot: All-In-One Controllable Conversational Model. In *Thirty-Fifth 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, 16081–16083. https://ojs.aaai.org/index.php/AAAI/article/view/18018

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.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021c. 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

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2010. TESLA: Translation evaluation of sentences with linear-programming-based analysis. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. 354–359.

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. 25–32.

Danyang Liu, Juntao Li, Meng-Hsuan Yu, Ziming Huang, Gongshen Liu, Dongyan Zhao, and Rui Yan. 2020b. A character-centric neural model for automated story generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1725–1732.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, et al. 2020d. Glge: A new general language generation evaluation benchmark. *arXiv preprint arXiv:2011.11928* (2020).

Feng Liu, Qirong Mao, Liangjun Wang, Nelson Ruwa, Jianping Gou, and Yongzhao Zhan. 2019a. An emotion-based responding model for natural language conversation. *World Wide Web* 22, 2 (2019), 843–861. https://doi.org/10.1007/s11280-018-0601-2

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021d. 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

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

Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020c. 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

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. Multilingual Denoising Pre-training for Neural Machine Translation. *CoRR* abs/2001.08210 (2020). arXiv:2001.08210 https://arxiv.org/abs/2001.08210

Ye Liu, Wolfgang Maier, Wolfgang Minker, and Stefan Ultes. 2021b. Naturalness Evaluation of Natural Language Generation in Task-oriented Dialogues using BERT. *arXiv preprint arXiv:2109.02938* (2021).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

Chi-kiu Lo. 2017. MEANT 2.0: Accurate semantic MT evaluation for any output language. In *Proceedings of the second conference on machine translation*. 589–597.

Chi-kiu Lo and Dekai Wu. 2011. MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 220–229.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. *Advances in Neural Information Processing Systems* 31 (2018).

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)*. 1116–1126.

Nurul Lubis, Sakriani Sakti, Koichiro Yoshino, and Satoshi Nakamura. 2018. Eliciting Positive Emotion through Affect-Sensitive Dialogue Response Generation: A Neural Network Approach. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 5293–5300. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16317

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*. 6020–6026.

Yubo Luo, Yongfeng Huang, Fufang Li, and Chinchen Chang. 2016. Text steganography based on ci-poetry generation using Markov chain model. *KSII Transactions on Internet and Information Systems (TIIS)* 10, 9 (2016), 4568–4584.

Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. *arXiv preprint arXiv:1910.03487* (2019).

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

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training Millions of Personalized Dialogue Agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 2775–2779. https://doi.org/10.18653/v1/d18-1298

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730* (2018).

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Efficient Estimation of Word Representations in Vector Space. (2013).

Tanushree Mitra, Clayton 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*. 1345–1354.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics* 6 (2018), 373–389.

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)*. 72–78.

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*. 311–318.

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

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*. 43–49.

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

Shrimai Prabhumoye, Alan W. Black, and Ruslan Salakhutdinov. 2020. Exploring Controllable Text Generation Techniques. *CoRR* abs/2005.01822 (2020). arXiv:2005.01822 https://arxiv.org/abs/2005.01822

Shrimai Prabhumoye, Khyathi Raghavi Chandu, Ruslan Salakhutdinov, and Alan W Black. 2019. " My Way of Telling a Story": Persona based Grounded Story Generation. *arXiv preprint arXiv:1906.06401* (2019).

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.

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.

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).

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. *arXiv preprint arXiv:1906.02738* (2019).

D. Radev, R. Zhang, A. Rau, A. Sivaprasad, C. Hsieh, N. F. Rajani, X. Tang, A. Vyas, N. Verma, and P. Krishna. 2020. DART: Open-Domain Structured Data Record to Text Generation. (2020).

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *CoRR* abs/1910.10683 (2019). arXiv:1910.10683 http://arxiv.org/abs/1910.10683

Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating Pretrained Language Models for Graph-to-Text Generation. *CoRR* abs/2007.08426 (2020). arXiv:2007.08426 https://arxiv.org/abs/2007.08426

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

Yu-Ping Ruan and Zhenhua Ling. 2021. Emotion-Regularized Conditional Variational Autoencoder for Emotional Response Generation. *IEEE Transactions on Affective Computing* (2021).

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40, 2 (2000), 99–121.

Bidisha Samanta, Mohit Agarwal, and Niloy Ganguly. 2020. Fine-grained Sentiment Controlled Text Generation. *arXiv preprint arXiv:2006.09891* (2020).

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. Discriminative Adversarial Search for Abstractive Summarization. *CoRR* abs/2002.10375 (2020). arXiv:2002.10375 https://arxiv.org/abs/2002.10375

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696* (2020).

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*. 35–40.

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

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

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 *Thirty-Fifth 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, 13798–13805. https://ojs.aaai.org/index.php/AAAI/article/view/17626

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.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. RUSE: Regressor Using Sentence Embeddings for Automatic Machine Translation Evaluation. In *WMT*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. *CoRR* abs/2010.15980 (2020). arXiv:2010.15980 https://arxiv.org/abs/2010.15980

Farhad Bin Siddique, Onno Kampman, Yang Yang, Anik Dey, and Pascale Fung. 2017. Zara returns: Improved personality induction and adaptation by an empathetic virtual agent. In *Proceedings of ACL 2017, system demonstrations*. 121–126.

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* (Montreal, Canada) *(NIPS'15)*. MIT Press, Cambridge, MA, USA, 3483–3491.

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

Haoyu Song, Yan Wang, Weinan Zhang, Xiaojiang Liu, and Ting Liu. 2020b. 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

Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020a. 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

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

Miloš Stanojević and Khalil Sima'an. 2014. BEER: BEtter Evaluation as Ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Baltimore, Maryland, USA, 414–419. https://doi.org/10.3115/v1/W14-3354

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize from human feedback. *CoRR* abs/2009.01325 (2020). arXiv:2009.01325 https://arxiv.org/abs/2009.01325

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

Bakhtiyar Syed, Gaurav Verma, Balaji Vasan Srinivasan, Anandhavelu Natarajan, and Vasudeva Varma. 2020. Adapting language models for non-parallel author-stylized rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9008–9015.

Pradyumna Tambwekar, Murtaza Dhuliawala, Animesh Mehta, Lara J. Martin, Brent Harrison, and Mark O. Riedl. 2018. Controllable Neural Story Generation via Reinforcement Learning. *CoRR* abs/1809.10736 (2018). arXiv:1809.10736 http://arxiv.org/abs/1809.10736

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)*. 5090–5099.

Alexey Tikhonov and Ivan P Yamshchikov. 2018. Guess who? Multilingual approach for the automated generation of author-stylized poetry. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 787–794.

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

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2020. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language* (2020), 101151.

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.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).

Ke Wang and Xiaojun Wan. 2018. SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks.. In *IJCAI*. 4446–4452.

Ruize Wang, Zhongyu Wei, Ying Cheng, Piji Li, Haijun Shan, Ji Zhang, Qi Zhang, and Xuanjing Huang. 2019b. Keep it consistent: Topic-aware storytelling from an image stream via iterative multi-agent communication. *arXiv preprint arXiv:1911.04192* (2019).

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019a. 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)*. 166–177.

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

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

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*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149* (2019).

Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, et al. 2020. A controllable model of grounded response generation. *arXiv preprint arXiv:2005.00613* (2020).

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

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. *CoRR* abs/2010.00840 (2020). arXiv:2010.00840 https://arxiv.org/abs/2010.00840

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled Text Generation With Future Discriminators. *CoRR* abs/2104.05218 (2021). arXiv:2104.05218 https://arxiv.org/abs/2104.05218

Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. 2019b. Enhancing topic-to-essay generation with external commonsense knowledge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2002–2012.

Xiaopeng Yang, Xiaowen Lin, Shunda Suo, and Ming Li. 2017. Generating Thematic Chinese Poetry with Conditional Variational Autoencoder. *CoRR* abs/1711.07632 (2017). arXiv:1711.07632 http://arxiv.org/abs/1711.07632

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019a. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *CoRR* abs/1906.08237 (2019). arXiv:1906.08237 http://arxiv.org/abs/1906.08237

Li Yujian and Liu Bo. 2007. A normalized Levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence* 29, 6 (2007), 1091–1095.

Y. Zeldes, D. Padnos, O. Sharir, and B. Peleg. 2020. Technical Report: Auxiliary Tuning and its Application to Conditional Text Generation. (2020).

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

Rui Zhang, Zhenyu Wang, and Dongcheng Mai. 2017. Building emotional conversation systems using multi-task Seq2Seq learning. In *National CCF Conference on Natural Language Processing and Chinese Computing*. Springer, 612–621.

Rui Zhang, Zhenyu Wang, Kai Yin, and Zhenhua Huang. 2019b. Emotional text generation based on cross-domain sentiment transfer. *IEEE Access* 7 (2019), 100081–100089.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. 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

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018b. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243* (2018).

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. 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

Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 670–680.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020c. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *ACL, system* *demonstration.*

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. 2020d. POINTER: Constrained Text Generation via Insertion-based Generative Pre-training. *CoRR* abs/2005.00558 (2020). arXiv:2005.00558 https://arxiv.org/abs/2005.00558

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019a. ERNIE: Enhanced Language Representation with Informative Entities. *CoRR* abs/1905.07129 (2019). arXiv:1905.07129 http://arxiv.org/abs/1905.07129

Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, YuSheng Su, Haozhe Ji, Jian Guan, Fanchao Qi, Xiaozhi Wang, Yanan Zheng, Guoyang Zeng, Huanqi Cao, Shengqi Chen, Daixuan Li, Zhenbo Sun, Zhiyuan Liu, Minlie Huang, Wentao Han, Jie Tang, Juanzi Li, Xiaoyan Zhu, and Maosong Sun. 2020a. CPM: A Large-scale Generative Chinese Pre-trained Language Model. *CoRR* abs/2012.00413 (2020). arXiv:2012.00413 https://arxiv.org/abs/2012.00413

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

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

Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. 2020. A Pre-Training Based Personalized Dialogue Generation Model with Persona-Sparse Data. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth 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

Peixiang Zhong, Chen Zhang, Hao Wang, Yong Liu, and Chunyan Miao. 2020. Towards persona-based empathetic conversational models. *arXiv preprint arXiv:2004.12316* (2020).

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 730–739. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16455

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.

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*. 1097–1100.

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

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