Why is constrained neural language generation particularly challenging?

Anonymous authors
Paper under double-blind review

Abstract

Recent advances in deep neural language models combined with the capacity of large scale datasets have accelerated the development of natural language generation systems that produce fluent and coherent texts (to various degrees of success) in a multitude of tasks and application contexts. However, controlling the output of these models for desired user and task needs is still an open challenge. This is crucial not only to customizing the content and style of the generated language, but also to their safe and reliable deployment in the real world. We present an extensive survey on the emerging topic of *constrained* neural language generation in which we formally define and categorize the problems of natural language generation by distinguishing between *conditions* and *constraints* (the latter being testable conditions on the output text instead of the input), present constrained text generation tasks, and review existing methods and evaluation metrics for constrained text generation. Our aim is to highlight recent progress and trends in this emerging field, informing on the most promising directions and limitations towards advancing the state-of-the-art of constrained neural language generation research.

1 Introduction

Recent advances in the field of natural language generation (NLG) Gatt & Krahmer (2018) have resulted in models able to produce realistic, coherent, and fluent texts in a multitude of natural language processing tasks. Powerful large scale language models can be readily used to perform unconditional language generation, however these models provide little control over attributes of the *generated* texts. Unlike conventional methods which were able to provide fine-grained control over many aspects of the system output including incorporating domain-specific dictionaries, terminology or certain words in the generated output, neural end-to-end approaches remove many of these knobs and switches Post & Vilar (2018). However, imposing constraints on the output generated by these models is crucial for achieving useful and safe language generation in a multitude of real world application scenarios. For example, it can help avoid generic and meaningless responses in dialogue systems See et al. (2019), personalize dialogue agents based on user features that lead to more engaging and meaningful conversations Zhang et al. (2018a), ensure non-offensive sentence completion and friendly communication Liu et al. (2019b), intervene on the system output in interactive scenarios where domain specific terminology must be included in the generated texts Crego et al. (2016), or aid in creative applications such as poetry generation or assisted story writing Peng et al. (2018). Moreover, controlling a generic pretrained language model in order to satisfy certain desiderata helps avoid generating toxic content, prevents demographic biases, can steer generations towards a desired topic or style Khalifa et al. (2020), and helps communicate intentions in suitable manners for different situations, target audiences and environments Lample et al. (2018), Li et al. (2018). Incorporating prior knowledge and target side constraints in text generative models has numerous applications in many natural language processing areas, including dialogue systems, machine translation, question answering, text summarization, text simplification, image captioning, etc. Unquestionably, constrained text generation is important in many real-world applications, but compared to other instances of natural language generation, constrained text generation using neural networks remains an open challenge.

We identify the following reasons that explain why constrained neural text generation represents a much harder problem compared to other instances of neural text generation: *i) lack of model expressiveness*: current models are not expressive enough to incorporate arbitrary constraints, defined as testable conditions on the *output* text, into the objective function at training time; *ii) lack of suitable evaluation metrics*: while one can verify whether an output satisfies a constraint or not, it is usually hard to measure *to what extent* an output satisfies a constraint, and it is even harder to jointly evaluate this with other properties of the generated text (such as relevance or coherence); *iii) difficulty in constrained optimization*: even if constraints can be expressed and added to the objective function, they are usually non-differentiable, especially at the token level. This is bad as most methods model and generate text as a sequence of tokens; *iv) lack of constrained text generation datasets* that are diverse and representative enough of the variety of practical constraints.

For example, commonly used sequential text generation methods and architectures assume a rigid modeling of the output sequence based on an ordering of words, in which tokens are generated progressively one at a time in a standard left-to-right manner Chan et al. (2019). Such autoregressive models cannot easily express constraints at arbitrary positions in the generated sequence or satisfy constraints involving multiple input objects. In addition to these issues, it is generally more challenging to incorporate multiple and heterogeneous constraints, which conform to given rules, topics, sentiments, lexical constraints, or pre-defined stylistic and content attributes.

Our work focuses on the emerging problem of neural natural language generation with constraints. We first define the problem and differentiate between the ambiguous use of *conditions* and *constraints* in natural language generation, including examples that represent instantiations of the constrained neural text generation problem. We then survey approaches, learning methodologies and model architectures employed for generating texts with desirable attributes, and corresponding evaluation metrics. We conclude with open research problems and limitations of current models. The scope of our work is draw clear boundaries between the confusing terminology used in the neural language generation literature, highlight the main approaches and discuss how they suffer from the general challenges of constrained text generation, and serve as an informative guide and an advocate for solving these general challenges and advancing meaningful, useful, and safe constrained NLG research.

2 Problem Definitions

We formally define the problem of natural language generation, accounting for context, conditions, and constraints placed on text generative models. First, we aim to articulate the key difference between condition and constraint since the distinction between these concepts is rather blurred in the natural language processing literature. Given a text generation task defined as $g(X) \rightarrow X'$, we define *condition* as a testable statement of the *input* X, and *constraint* as a testable statement of the *output* X'.

Accounting for the distinction above, we divide the text generation problem into three categories: *i) generic or free-text generation* which we present in Section 2.1, *ii) conditional text generation* which we introduce in Section 2.2, and *iii) constrained text generation* which we outline in Section 2.3. The focus of our work is on the particular problem of *constrained* text-to-text generation, leaving aside text generation tasks from other types of inputs such as data-to-text generation or image-to-text generation which are *conditional* in nature according to our definitions.

2.1 Generic/Free-Text Generation

The problem of generic text generation considers the intrinsic history of words generated until the current timestep in the sequence as context, and does not place any external user-defined conditions or constraints on the model output.

Given a discrete sequence of text tokens $x = (x_1, x_2, \dots, x_n)$ as input where each x_i is drawn from a fixed set of symbols, generic text generation aims to learn the unconditional probability distribution $p(x)$ of sequence x. This distribution can be auto-regressively factorized using the chain rule of probability Bengio et al. (2003) into a product of conditional probabilities $p(x) = \prod_{i=1}^{n} p(x_i|x_{<i})$ to perform density estimation and

generation of language data. When $p(x)$ is modeled by a neural network with parameters θ, the neural network is trained to minimize the negative log-likelihood $\mathcal{L}(D) = -\sum_{k=1}^{|D|} \log p_\theta(x_i^k | x_{<i}^k)$ over a collection of samples $D = \{x^1, \ldots, x^{|D|}\}$. To generate new samples, each token x_i is iteratively sampled from $p_\theta(x_i | x_{<i})$ and is fed back into the model as the input for the next timestep.

Large scale models for generic text generation show promising abilities to imitate the distribution of natural language and generate long-term realistic and coherent texts, however such free-text generation models place a lot of burden on the generative model to capture complex semantic and structural features underlying the data distribution; this can often result in repetitive, contradictory, and largely randomized generated texts Holtzman et al. (2019). Notably, the content generated by free-text generative models cannot be controlled with respect to particular attributes and modes of the data distribution. This inability to control which regions of the data distribution are generated is particularly problematic considering there is significant toxicity, hate, bias, and negativity present in the large-scale web crawled datasets text generation models are commonly trained on. Imposing conditions or constraints on the generation process results in safer and more useful generated texts for downstream application tasks Krause et al. (2020).

2.2 Conditional Text Generation

Conditional text generation manipulates attributes of the generated content depending on specific contexts or user needs, and allows the data generation process to focus on specific modes of the data. Conditioning the generative model on additional information makes it possible to generate texts which satisfy given *input* conditions and meet desired attributes. In the literature conditional text generation is sometimes referred to as context-dependent text generation. While the word context may carry different semantics for different readers, in this survey we consider as context only attributes which are inherently external to the model itself; model intrinsic attributes such as for example, the history of past generated words, is already included in the formulation of generic text generation. For example, context attributes used for conditioning generated texts are the *source sentence* in machine translation, the *conversational history* in dialogue systems, the *input document* in text summarization and text simplification, the *input question* in question answering systems, or contextual information such as *product, time*, and *location* in review generation.

Conditional text generation models add a contextual variable or attribute code c to the probabilistic model $p(x)$ transforming it into a conditional probability model $p(x|c)$, which can be auto-regressively decomposed using the chain rule of probability $p(x|c) = \prod_{i=1}^{n} p(x_i | x_{<i}, c)$. When $p(x|c)$ is modeled by a neural network with parameters θ, the model minimizes the negative log-likelihood loss function accounting for the attribute code c: $\mathcal{L}(D) = -\sum_{k=1}^{|D|} \log p_\theta(x_i^k | x_{<i}^k, c^k)$. Besides generation, conditional models can also be used as generative classifiers to compute $p(c|x_{<i})$ by applying Bayes rule.

2.3 Constrained Text Generation

The problem of constrained text generation is focusing on generating coherent and logical texts that do (not) cover lexical concepts (for eg., pre-defined nouns, verbs, entities, phrases or sentence fragments) desired to be (not) present in the *output*, as well as generate outputs that abide to specific format, semantic, syntactic or utility rules to reflect the particular interests of the system user. Constraints impose restrictions on the generative model that must be satisfied by any solution to the optimization problem and their fulfillment can be tested accordingly. In the literature the distinction between conditional, controlled, and constrained text generation is not clearly defined, and these terms are often used interchangeably. In fact, the first work that proposed generating constrained text is actually referring to the task as "controlled" generation Hu et al. (2017). In what follows we formally define the problem of constrained text generation.

Let us consider we are (optionally) given an unordered or ordered set of n concepts $x = \{c_1, c_2, \ldots, c_n\} \in \mathcal{X}$, where $\mathcal{X}$ denotes the space of all concepts, and $c_i \in C$ is a concept belonging to the concept vocabulary C. In addition, let us assume we are also (optionally) given a set of m rules $y = \{y_1, y_2, \ldots, y_m\} \in \mathcal{Y}$, with $y_i \in \mathcal{R}$, where $\mathcal{R}$ denotes the space of all rules, and each y_i is a text generation constraint expressed in logical form. We formulate constrained text generation as learning the structured predictive function $f : \mathcal{X} \cup \mathcal{Y} \to \mathcal{Z}$, where $\mathcal{X} \cup \mathcal{Y} \neq \phi$ which maps a set of concepts and/or constraint rules to a generated sentence. Therefore, constrained text generation methods impose constraints on the generated sentences and produce output in the form of grammatical sentence $z \in \mathcal{Z}$ which contains all concepts present in x and all constraint rules specified in y. The probability $p(z|f)$ can still be modeled autoregressively $p(z|f) = \prod_{i=1}^{n} p(z_i|z_{<i}, f)$; when $p(z|f)$ is modeled by a neural network with parameters θ, the negative log likelihood function can be minimized while leveraging f for constraint satisfaction $\mathcal{L}(D) = -\sum_{k=1}^{|D|} \log p_\theta(z_i^k|z_{<i}^k, f)$.

The matching function f manipulates the probability distribution and indicates to which extent the constraints are satisfied. In the literature, constrained text generation methods can be either *i) Soft-constrained (priming)*, when the matching function f is a soft measure of semantic similarity and only requires the generated sentences to be semantically related to the given constraints, or *ii) Hard-constrained*, when the matching function f is a binary indicator which rules out the possibility of generating infeasible sentences that do not meet the given constraints. Hard-constrained text generation is notably a more challenging task compared to soft-constrained text generation, and it requires designing specialized approaches and architectures to ensure the constraints in the output sentence. In contrast, soft-constrained text generation models are usually easier to design, e.g., with the use of existing copy and attention mechanisms for soft enforcing constraints and annotated keyword-text pairs; nevertheless, some of these soft constraints are likely to be lost during generation, especially if multiple weakly correlated (lexical) constraints must be included Zhang et al. (2020).

Compared to generic text generation which assumes no conditions on input or output other than existing context, and compared to conditional text generation which places conditions on the input which can be considered at training time, constrained text generation places conditions on the output which is a considerably more difficult and challenging problem to solve. Unlike input conditions, output conditions cannot be considered at training time and their satisfaction is assessed after training has completed by sampling and inspecting the generated outputs. In addition, standard sequence generation architectures are not designed to easily accommodate or incorporate output constraints. Given the model structure itself cannot express output conditions, it becomes challenging to evaluate the extent to which constraints are satisfied by a model, objectively compare and contrast the performance of different models, and measure overall success to inform on progress in constrained natural language generation. Due to these limitations, current methods proposed to address constrained text generation are neither satisfactory nor sufficient. The main machine learning challenge is that it is hard to evaluate the objective function for constrained text generation, and very few works have approached the problem from the prism of editing the objective function to incorporate constraints at training time. Even if constraints were to be added to the objective function itself, constrained optimization would be another challenge. In general, reinforcement learning approaches are used in the context of text generation to optimize non-differentiable reward functions computed at the token level, for eg., BLEU in machine translation or ROUGE in text summarization. However, optimizing such automatic measures that focus on local n-gram patterns often results in deteriorated textual outputs despite increased automatic scores Bosselut et al. (2018), Pasunuru et al. (2020). Moreover, applying reinforcement learning to text generation at the word level leads to difficulty in proper temporal credit assignment for long-term textual rewards Saleh et al. (2019). Given that the environment provides only delayed rewards as the agent executes a sequence of actions, it is impossible to know whether the agent succeeds in achieving a task until the end of the episode, at which point the agent needs to determine which of the actions in the sequence are to be credited with producing the resulting reward Gao et al. (2019a). Adding constraints on top of existing reinforcement learning issues would be detrimental to the learning process, if not make learning close to impossible: the objective function would be even harder to optimize, rewards would be delayed, sparse and non-informative. Despite these open problems and limitations, we argue neural constrained text generation is an important research area which deserves a lot more attention.

Constrained text generation is useful in many scenarios, such as incorporating in-domain terminology in machine translation Post & Vilar (2018), improving semantic corectness Balakrishnan et al. (2019), avoiding generic and meaningless responses in dialogue systems using grounding facts Mou et al. (2016), paraphrase generation in monolingual text rewriting Hu et al. (2019), Kajiwara (2019), incorporating ground-truth text fragments (such as semantic attributes, object annotations) in image caption generation Anderson et al. (2017), creating a story Fan et al. (2018b) or poem Ghazvininejad et al. (2017) using a pre-defined set of keywords, or re-writing a user search query as a fluent sentence. Typical attributes used to generate constrained natural language are the *tense* and the *length* of the summaries in text summarization Fan et al. (2018a), the *sentiment* of the generated content in review generation Mueller et al. (2017), *language complexity* in text simplification or the *style* in text style transfer applications. In addition, constrained text generation is used to overcome limitations of neural text generation models for dialogue such as genericness and repetitiveness of responses See et al. (2019), Serban et al. (2016).

Nevertheless, generating text under specific lexical constraints is challenging. Common models and architectures employed for natural language generation are autoregressive in nature, generating tokens one by one in a sequential manner from left to right; by design, these models lack fine control over the generated sequence and cannot easily support constraints at arbitrary positions in the output or constraints involving multiple input objects Zhang et al. (2020), Hsieh et al. (2021). While for humans it is straightforward to generate sentences that cover a given set of concepts or abide to pre-defined rules by making use of their commonsense reasoning ability, generative commonsense reasoning with a constrained text generation task is more challenging for machine learning models Lin et al. (2019).

3 NLG Constraints

Natural language generation models place restrictions on the generated output to produce texts that reflect certain user preferences. In Table 1 we present NLG tasks distinguishing between conditions and constraints. We broadly group existing constraints into the following categories:

Table 1: Overview of constrained NLG tasks, differentiating between conditions and constraints.

Task	**Condition**	**Constraint**				
		Lexical	*Format*	*Semantic*	*Syntactic*	*Utility*
Machine Translation	source input	words phrases entities	–	topic sentiment	paraphrase tense gender pronouns	target language politeness factuality/faithfulness helpfulness, harmlessness
Dialogue Generation	past utterance(s)	words phrases entities	length verbosity	topic sentiment toxicity	paraphrase gender pronouns	politeness personality traits factuality/faithfulness helpfulness, harmlessness
Text Summarization	input document(s)	words phrases entities	length	topic	paraphrase	factuality/faithfulness helpfulness, harmlessness
Text Simplification	input text	words phrases entities	length	topic	paraphrase	simpler vocabulary readability factuality/faithfulness helpfulness, harmlessness
Text Style Transfer	source text	words phrases entities	length	topic sentiment	paraphrase tense gender pronouns	style factuality/faithfulness
Question Answering	input question	words phrases entities	length	topic	paraphrase tense gender pronouns	factuality/faithfulness politeness helpfulness, harmlessness
Narrative Generation/ Story telling	–	words phrases entities entities	length	topic sentiment	paraphrase tense gender pronouns	readability factuality/faithfulness helpfulness, harmlessness style
Poetry Generation	–	words phrases entities	length rhyme rhythm	topic sentiment	paraphrase tense gender pronouns	readability factuality/faithfulness style

Lexical constraints Lexical constraints serve with the inclusion of specific keywords, phrases or entities at arbitrary positions in the output, and can be specified as a word (a single token) or phrasal constraint (a multi-word phrase). They are useful in tasks such as *dialogue generation*, *machine translation*, *story telling* or *poetry generation*.

Format constraints Format constraints such as number of sentences, length of sentences, order of words, number of syllables, etc. serve to denote preferences on the form and appearance of the generated output. Format constraints are particularly useful in tasks such as *poetry generation* to specify the form of the generated poem, for eg. quatrain or regulated verse, length of the poem, rhyme and rhythm. In *text summarization* or *text simplification*, length constraints define the length of the generated output to be strictly less than the length of the input document, while in *dialogue generation* they help define the level of verbosity of the dialogue agent.

Semantic constraints Semantic constraints are used to define the topic and sentiment of the generated content, or control fine-grained aspects such as removing toxicity. Topic constraints are particularly useful in *dialogue generation*, where the goal is to generate on-topic responses that are safe, non-harmful, unbiased, relevant to the dialogue context and particular user needs; in *story telling* or *poetry generation*, topic constraints help define the theme. Generating language that conveys particular positive, neutral or negative sentiment aims to endow artificial agents with human-like traits such as compassion, empathy, and enables agents to react with appropriate emotion in diverse social situations; constraining on a specific sentiment is important in many tasks such as *dialogue generation*, *review generation*, *story telling*, *poetry generation* or *text style transfer*. Furthermore, increasing politeness of a dialogue system or reducing toxicity of generated language are important aspects with respect to human-centered metrics of conversation quality.

Syntactic constraints Syntactically constrained text generation produces sentences with desired syntax by incorporating syntactic templates and rules in the training of the text generative model. Syntactic constraints are useful in paraphrase generation, where given a sentence and a target syntactic form (e.g., a constituency parse), a system must produce a paraphrase of the sentence whose syntax conforms to the target Iyyer et al. (2018). Generating texts that convey the same meaning but with different expressions has numerous applications in many natural language generation tasks, including monolingual transduction tasks such as *text simplification*, *text compression*, or *text style transfer*, as well as in tasks like *text summarization*, *machine translation* or *question answering* where alternative ways of expressing the same information capture the inherent language variations.

Utility constraints Utility constraints capture holistic properties of the generated output, for eg., stylistic, readability, faithfulness, helpfulness, harmlessness or politeness aspects. Preserving the information content of texts while manipulating attributes such as style, readability level, personality traits of the user or specific gender pronouns allows to customize generated texts to different audiences and make them relevant in a wide variety of end-user applications. Stylistic constraints are immediately relevant to the task of *text style transfer*, with applicability in many tasks, including *dialogue generation*, *machine translation*, *text simplification*, *story telling*, *poetry generation*, *review generation*.

Constraining text generation on attributes such as readability and level of text complexity serves to adapt the generated output to users of different age, backgrounds and educational levels. Reducing complexity of texts while preserving the information content is the main goal of *text simplification*; in addition, in tasks such as *dialogue generation*, *text summarization*, *story telling*, *poetry generation*, *question answering* it is important to customize texts for various literacy levels.

In many languages the degree of politeness is an important aspect of inter-personal communication, and honorifics are used to express courtesy, social distance, or the relative social status between the speaker and their addressee(s) Sennrich et al. (2016). Politeness constraints on the output are used in *machine translation*, *dialogue generation*, *story telling*, and *text style transfer*.

Faithfulness constraints enforce similarity between a generated text sequence and its corresponding input, requiring models to generate texts that are faithful, factual and preserve the original information content.

Such constraints are important in many tasks, including *text summarization*, *machine translation*, *text simplification* or *dialogue generation*, where models are vulnerable to producing hallucinated content.

Language constraints are useful when translating texts between different languages such as in *machine translation*, or from complex language into simple language such as in *text simplification*.

4 Constrained Natural Language Tasks

In what follows we briefly describe NLG tasks, differentiating between conditions and constraints.

Machine Translation Machine translation is focusing on the automatic translation of textual content from one language into another language, and is a typical example of both *conditional and constrained text generation*, as it conditions on the input text in the source language and constraints the model to generate fluent and faithful output in the target language. Additional constraints can be placed on the degree of formality and politeness, the use of gender-specific pronouns, the inclusion in the target sentence of named entities or specific concepts from the source sentence.

Dialogue Systems A dialogue system, also known as a conversational agent, is a computer system designed to converse with humans using natural language. Dialogue generation is an instance of *conditional text generation* where the system response is conditioned on the previous user utterance and frequently on the overall conversational context. Dialogue generation can also be an instance of *constrained text generation* - it is desirable generated dialogues incorporate explicit personality traits Zheng et al. (2019), control the sentiment Kong et al. (2019), topic, degree of formality and politeness of the generated response to resemble human-to-human conversations. In addition, dialogue responses may need to incorporate text excerpts from past dialogue history or entities such as locations, persons, institutions, etc. From an application point of view, dialogue systems can be categorized into: *i) task-oriented dialogue agents*, designed to help users complete a particular task, or *ii) non-task oriented dialogue agents (chat-bots)* designed to carry entertaining conversations with their users on a wide range of open domains. A common problem in dialogue generation systems is that they tend to generate safe, universally relevant responses that carry little meaning Serban et al. (2016), Li et al. (2016a), Mou et al. (2016). Moreover, they can fail to take turns asking questions and balance specificity with genericness of the output See et al. (2019).

Text Summarization Text summarization facilitates a quick grasp of the essence of a document and produces a condensed version of its content, by copy-pasting the relevant portions from the input as in extractive summarization Nallapati et al. (2017), or by generating novel content as in abstractive summarization Rush et al. (2015), Nallapati et al. (2016), See et al. (2017), or via hybrid approaches Liu et al. (2018) that combine both techniques. Text summarization is a *conditional text generation task* where the condition is represented by the given document(s); additional conditions are used in remainder summarization to flexibly define which parts of the document(s) are of interest, for eg., remaining paragraphs the user has not read yet, or in source-specific summarization to condition summaries on the specific input source and style of writing, for eg., newspapers, books or news articles. Text summarization is also a *constrained text generation* task considering that the length of the summary is fixed, pre-determined, and strictly less than the original document; this allows to digest information at different levels of granularity and detail according to user needs and time budgets. Moreover, constraints can be placed on specific concepts to include in the summary, such as named entities, or on explicitly picking sentences from the original document as in extractive summarization.

Text Simplification Text simplification is designed to reduce the text complexity, while preserving its original meaning. In the literature, simplification has been addressed at multiple levels: *i) lexical simplification* focused on replacing complex words or phrases with simpler alternatives; *ii) syntactic simplification* alters the syntactic structure of the sentence; *iii) semantic simplification* paraphrases portions of the text into simpler and clearer variants. End-to-end models attempt to combine all these steps. Text simplification is both *conditional and constrained text generation*; we are conditioning on the input complex text to generate a simpler version, accounting for constraints such as higher readability, simpler vocabulary, and shorter sentence length than the complex input.

Text Style Transfer Style transfer has its origins in computer vision applications for image-to-image translation and more recently has been used in natural language processing applications for machine translation, sentiment modification to change the sentiment of a sentence from positive to negative and vice versa, word substitution decipherment and word order recovery Hu et al. (2017). Text style transfer is designed to preserve the information content of a source sentence while altering the way it is delivered to meet desired presentation constraints. Textual content is disentangled from the style in which it is presented, and manipulating stylistic attributes can be done without parallel aligned data between source and target styles. Text style transfer is an instance of both *conditional and constrained text generation* given that we condition on the given source text and constrain the transferred sentences to stylistically match target examples.

Question Answering Question answering systems are designed to find and integrate information from various sources to provide responses to user questions Fu & Feng (2018). While traditionally candidate answers consist of words, phrases or sentence snippets retrieved and ranked appropriately from knowledge bases and textual documents Kratzwald et al. (2019), answer generation aims to produce more natural answers by using neural models to generate the answer sentence. Question answering is both *conditional and constrained text generation* task; the system conditions on the user question, and simultaneously ensures that concepts needed to answer the question are present in the generated output. Diverse question answering systems are proposed in the literature addressing for eg., medical information needs Wiese et al. (2017), mathematical questions Schubotz et al. (2018), quiz bowl questions Iyyer et al. (2014), cross-lingual and multi-lingual questions Loginova et al. (2018). Notably, in practical applications users are not only interested in learning the exact answer word or phrase, but also in how it relates to background information and to previously asked questions and answers Fu & Feng (2018).

Narrative Generation/Story Telling Neural narrative generation is an important step towards computational creativity Gervás (2009) and represents a long-form open-ended text generation task which simultaneously addresses the selection of appropriate content (*"what to say"*) and the surface realization of the generation (*"how to say it"*)Wiseman et al. (2017). Narrative generation is a *constrained text generation* task that places explicit constraints on concepts to steer the narrative in particular topic directions and expands the few keywords specified as the story title, beginning or ending. While existing models can generate stories with good local coherence, generating long stories is challenging. Difficulties in coalescing individual phrases into coherent plots and in maintaining character consistency throughout the story lead to a rapid decrease in coherence as the output length increases van Stegeren & Theune (2019). Hierarchical models for story generation break down the generation process into multiple steps: first modelling the action sequence, then the story narrative, and finally entities such as story characters Fan et al. (2019). Neural narrative generation combining story-writing with human collaboration in an interactive way improves both story quality and human engagement Goldfarb-Tarrant et al. (2019).

Poetry Generation The poem generator operates in an interactive context where the user supplies the model with a set of ordered concepts that reflect her writing intent, as well as the format of the poem, for eg. quatrain or regulated verse. Poetry generation is a *constrained text generation* problem since user defined concepts need to be included in the generated poem, and a *conditional text generation* problem given the explicit conditioning on stylistic attributes. For a detailed overview of poetry generation see Oliveira (2017).

5 Constrained NLG Methods

Accounting for the different types of constraints introduced in Section 3, we distinguish five methodologies commonly employed in the constrained text generation literature: *i)* decoding approaches, *ii)* fine-tuning approaches, *iii)* discriminative approaches, *iv)* edit-based approaches, and *v)* adapting existing models and architectures to accommodate constraints on the generated output. In what follows we present each approach in detail, outlining the main associated challenges.

5.1 Decoding approaches

The most popular approach to text generation in the literature has been supervised learning with task-specific datasets; nevertheless since many real-world applications require diverse and potentially evolving constraints, it is infeasible to annotate task-specific training data for every combination of constraints Qin et al. (2022). Even if collecting the data was not a bottleneck, re-training large language models that are extreme in scale for each new constraint or combination of constraints is undesirable. The alternative to fine-tuning language models with task-specific datasets is to enrich decoding algorithms so as to accommodate constraints on the fly. We present decoding approaches to constrained text generation below.

Lexical constraints ***Lexically constrained (guided) decoding*** aims to restrict the search space at decoding time to sequences which contain pre-defined lexical constraints only. These lexical constraints can be specified in the form of a word constraint (a single token) or a phrasal constraint (a multi-word phrase, i.e. a sequence of two or more contiguous tokens). To this end, the beam search decoding algorithm is modified to enforce the inclusion of pre-specified words and phrases in the generated output by allowing the model distribution to not only account for the given lexical constraints, but also to generate parts of the output sequence not covered by the constraints. In general, the decoder can more easily place multiple sequential tokens in a phrasal constraint (where the permutation order is fixed) on the generated output as opposed to placing multiple separate, independent constraints. In addition, the lexically constrained decoding approach assumes lexical constraints are pre-determined, which may not always be the case; if so, the open question is where to get lexical constraints from.

Early work on constrained decoding in machine translation relies on the ***placeholder approach*** designed to recognize identifiable elements (numbers and named entities) in the source sentence, temporarily replace these with corresponding placeholders during preprocessing, and then substitute the assigned placeholders with the original source-language strings during beam search decoding Crego et al. (2016). Nevertheless, such an approach is limited and unable to model the source tokens in target language specific terminology or the vocabulary from a new out-of-distribution domain. ***Prefix decoding*** represents a modification of beam search to first ensure that a user defined target prefix is generated first, and only after build hypotheses for the suffix that maximize the coverage of the remaining source-side tokens. As decoding progresses from left to right, the decoder transitions from a constrained prefix decoding mode to unconstrained beam search. For example, the start of the sentence symbol $<s>$ can be easily included as the first word of a constraint Knowles & Koehn (2016), Wuebker et al. (2016). In the context of text summarization, an essential property of a summarization system is the ability to generate a summary with desired length. ***Grid beam search*** Hokamp & Liu (2017) extends beam search decoding to allow for the inclusion of arbitrary target side hard lexical constraints at any position in the generated sequence. Given C input constraints, the algorithm maintains $C+1$ separate beams $B_0, B_1, \ldots, B_c$ that group together hypotheses which meet the same number of satisfied constraints. Decoding runs similar to beam search, with an additional dimension added to keep track of how many constraints are met by each hypothesis at every timestep; the highest scoring hypothesis in beam B_c is ultimately generated. However, grid beam search is impractical as decoding complexity is linear in the number of constraints, i.e. beam size increases proportionally to the amount of constraints and changes for every sentence. ***Constrained beam search*** Anderson et al. (2017) guarantees the inclusion of input constraints in the generated sentences by extending beam search with a finite state machine whose states mark completed subsets of the input set of constraints; however, decoding complexity has an exponential cost in the number of constraints, making it infeasible in many applications. ***Dynamic beam allocation*** Post & Vilar (2018) improves upon the runtime complexity of grid beam search and constrained beam search by decoding with constant complexity $O(1)$ in the number of constraints. The algorithm still groups together hypotheses that have met the same number of constraints by using a single fixed-size beam which is dynamically divided at each time-step according to how many constraints have been met. Despite being more efficient, dynamic beam allocation does not necessarily outperform conventional beam search Lin et al. (2019). In addition, the generation of hypotheses that only partially satisfy a phrasal constraint needs to be aborted to unwind to the tokens in the constraint. ***Neurologic decoding*** Lu et al. (2021) modifies beam search to enforce the satisfaction of lexical constraints expressed under predicate logic in conjunctive normal form (CNF). Given the intractability of exhaustive beam search to optimize CNF constraints, the algorithm searches for approximately-optimal output sequences in which all clauses are satisfied, including both positive and negative constraints (i.e. words that must be generated, respectively omitted in the output sequence). The method is applied to cooking recipe generation, where the task is to generate cooking instructions given a dish name and a list of ingredients, and to data-grounded dialogue response generation where a response is generated given a query and a list of facts to convey.

In general, lexically constrained decoding methods have high computational complexity and force the inclusion of specific words in the generated sentence at every timestep of the generation process with no prior examination of these specific words before generation begins Latif et al. (2020); this unnatural way of generating sentences can impact the quality and naturalness of the generated output Liu et al. (2019a), Post & Vilar (2018). In lack of suitable evaluation metrics, there is no commonly agreed criteria for objectively assessing the quality of the generated sentences and conducting comparisons across text generation models.

Format constraints ***Fixed length decoding*** Kikuchi et al. (2016) constrains the length of generated summaries in two ways: *i)* by preventing the decoder from generating the end-of-sentence tag until the length of the generated sequence exceeds the desired length, and *ii)* by defining the minimum and maximum length range of the sequence and discarding out-of-range sequences. ***Non-monotonic decoding*** approaches allow tokens to be inserted at any position in the generated sequence during decoding, therefore accommodating flexible orderings of the output. Unlike left-to-right autoregressive generation that produces a single word at a time, non-monotonic decoding can satisfy lexical constraints at multiple locations in the output sequence allowing for highly parallel generation and faster decoding times. Nevertheless, such approaches assume the generated sequence length is known a priori, preventing it from being dynamically adjusted as generation proceeds. Moreover, such models assume conditional independence between output tokens, i.e. tokens are generated independently, may be inconsistent and agnostic to each other. Consequently, this approach may hurt the expressiveness of the model and lead to potential performance degradation, impacting the fluency and naturalness of the output. In addition, non-monotonic sequence decoding approaches can terminate prematurely before constraints are satisfied in the output sequence Zhang et al. (2020), Hsieh et al. (2021). The main limitation of this approach is the lack of model expressiveness in accommodating constraints.

Insertion Transformer Stern et al. (2019) proposes a flexible sequence generation framework based on repeated insertion operations into an initially empty output sequence until a termination condition is met. The model adopts a progressive masking approach based on token importance in the original text and is trained to generate a missing token between every two tokens in the input. To this end, the original Transformer Vaswani et al. (2017) decoder is modified to allow insertions not just at the end but anywhere in the output sequence. The model can decode sequences serially one token at a time, or it can decode sequences in parallel with simultaneous insertions at multiple locations. A similar approach is considered in InDIGO Gu et al. (2019a) which extends Transformer for insertion-based decoding with inferred generation order. Token generation order for the output sequence is modeled as a latent variable, and at each decoding step the model predicts both the generated word and its position in the output sequence; nevertheless, strong conditional independence is assumed between the output tokens which hurts output quality. An iterative refinement step based on latent variables is added to the Transformer decoder to refine a target sequence gradually over multiple steps until a predefined stopping criterion is met Lee et al. (2018). Progressive Insertion Transformer Zhang et al. (2020) uses non-autoregressive modeling based on a top-down progressive structure for lexical hard-constrained text generation. Given lexical constraints as input, the model inserts tokens progressively according to word importance to generate the target sequence, as follows: first it generates high-level words in a sentence such as nouns, adjectives and verbs, then uses these as pivoting points to insert details of finer granularity and finally completes the sentence by adding connecting words which carry less information, such as pronouns and prepositions. Entity Constrained Insertion Transformer Hsieh et al. (2021) builds upon previous models considering hard lexical constraints in the form of entities in the output sequence. Similar approaches train the Transformer decoder to insert missing tokens in a partially complete sequence without relying on a pre-specified factorization of tokens Chan et al. (2019), Gu et al. (2019b); based on the information available in the sequence, the insertion-based generative model is able to dynamically infer the remaining parts irrespective of their arbitrary order.

Syntactic and Semantic constraints ***Distributional constraints*** Baheti et al. (2018) on topic and semantic similarity are used to incorporate source side-information at decoding time in neural conversational systems and encourage the generation of more diverse responses. Moreover, constraints over topics and syntax are used to generate matching or semantically similar statements in response to the user input Niu & Bansal (2018). Lexically constrained decoding from pre-trained language models aims to steer language models in useful and safe directions so as to minimize the risks associated with these models generating biased, offensive and toxic content Sheng et al. (2019), Holtzman et al. (2019). **Weighted decoding** methods employ a linear combination of output logits from multiple prompts (raw prompt vs prefix-prepended prompt) to vary the strength of desired target attributes in the output text Pei et al. (2023), Zhang & Song (2022). **Energy-based constrained decoding** allows the specification of style and lexical constraints through an energy function and performs differentiable reasoning through gradient-based sampling Qin et al. (2022), Mireshghallah et al. (2022). Constrained text generation is viewed as an optimization problem, where the goal is to iteratively seek text with lower energy. The sampling process uses gradients of the energy function to update a continuous relaxation of text data, which is then mapped back to the discrete space of natural language via a discretization approach. For streering the generation towards desired constraints, biases are applied to the logits of the pre-trained model output layer, which is also found to improve the speed of the decoding process Liu et al. (2023b). Nevertheless, sampling from energy-based models requires many iterations to converge to plausible text.

5.2 Fine-tuning approaches

Semantic and Utility constraints Controlling the output of pre-trained language models is crucial in a wide-range of safety-critical applications, including mental health support chatbots, sentiment controlled text generation, language detoxification, etc. To this end, fine-tuning approaches are used for fine-grained control over individual stylistic aspects (for eg., length, professional and descriptive style, tense, personal voice, gender) and content aspects (for eg., sentiment and topic) of the generated texts Ficler & Goldberg (2017), Lample et al. (2018). Typically, the pre-trained model is fine-tuned separately for each attribute of interest, which poses the challenge of how to learn disentangled latent representations of style and content in neural language models John et al. (2019) and isolate the desired attribute from the distribution shift between the generative model and the fine-tuned dataset. The lack of datasets that are diverse and representative of constrained criteria encountered in practice represents an open challenge for fine-tuning pre-trained models.

CTRL Keskar et al. (2019) uses control codes to generate texts that meet user-defined constraints on domain, style, topics, dates, entities, relationships between entities, plot points, and task-related behavior. These pre-defined codes are appended at the beginning of raw text sequences to define task-specific training data and create controllable task-specific behaviour at sampling time. Similarly, fine-grained semantic control codes are used to steer generation towards targeted attributes Ross et al. (2022). Decoding Experts (DExperts) Liu et al. (2021) is a decoding-time method for constrained text generation which combines a pre-trained language model with both an “expert” and “anti-expert” language model in a product of experts. The “expert” models desirable aspects of the generated text (for eg., positive sentiment), while the “anti-expert” plays the antagonistic role of modeling undesirable attributes to be avoided (for eg., toxicity); each one of the three language models is conditioned on the same user prompt. While the method highlights the promise of customizing decoding from pre-trained language models in safe and efficient ways, gathering large amounts of toxic data to model undesirable attributes may be challenging. In general, adding negativity to a positive prompt is a much easier task than adding a positive turn to a negative prompt Madotto et al. (2020). Aligning LLM models with user preferences at inference time is important for personalizing LLM models to their users without the need to retrain the model for each new target attribute of interest. SteerLLM Dong et al. (2023) allows end-users to customize responses during inference by conditioning a supervised finetuned model on a multi-dimensional set of user attributes.

Reinforcement learning from human feedback (RLHF) Christiano et al. (2017), Stiennon et al. (2020) has become a crucial component in improving the generation abilities of LLMs. Aligning AI models with human preferences is considered crucial for safely deploying artificial systems in the real-world and ensuring they exhibit behaviors consistent with human values Ouyang et al. (2022), OpenAI (2022), Ziegler et al. (2019), Shen et al. (2023), Wang et al. (2024b). RLHF algorithms further finetune under a KL-constrained RL objective a language model that has already undergone supervised fine-tuning (referred to as the reference model); this objective encourages the model to maximize the reward and simultaneously discourages high

KL divergence between the language model and the reference model. The reward model is derived from human preferences on text continuations with positive sentiment or vividly descriptive language, while the KL constraint is used to prevent the fine-tuned policy from drifting too far from the reference policy. In dialogue systems, KL control has been used to retain prior information and penalize divergence from the pre-trained model during RL fine-tuning Jaques et al. (2019). Despite the promise of RLHF in aligning LLMs to user preferences, fine-grained control over large language models remains a significant challenge Wang et al. (2024a). Since RLHF training can be difficult and unstable, alternative approaches finetune language models to mimic the Best-of-N (BoN) Nakano et al. (2021), Touvron et al. (2023) distribution by minimizing the KL divergence between the language model and the BoN distribution, for example when generating movie reviews with positive sentiment Amini et al. (2024) or in text summarization Gui et al. (2024). Nevertheless, KL does not capture attributes of generated text that humans judge to be salient, such as the length of the generated response. The lack of suitable evaluation metrics is an outstanding challenge in generating high quality outputs.

Pre-trained OpenAI-GPT2 Radford et al. (2019) model is used to re-write a story through counterfactual reasoning and generate a narrative consistent with the imposed constraints Qin et al. (2019). In abstractive summarization, OpenAI-GPT2 is used in a reinforcement learning setting which trains the summarization agent to maximize coverage and fluency of the generated content constrained on a pre-defined length Laban et al. (2020). RecipeGPT H. Lee et al. (2020) fine-tunes the GPT-2 pre-trained language model for generating cooking instructions when hard constraints are placed on the recipe title and ingredients; the model can also generate the list of ingredients for a recipe when constrained on the recipe title and specific cooking instructions. Infilling by language modeling is used to complete variable length text spans (e.g. words, n-grams and sentences) by fine-tuning a pre-trained language model on sentence pairs that contain both artificially-masked text and the corresponding original text Donahue et al. (2020).

While fine-tuning models on task specific datasets has become the dominant paradigm for constrained text generation from pre-trained large language models, these models generally fail to reliably incorporate the underlying constraints in the generated texts even when supervised with large amounts of task-specific examples Lu et al. (2021). Moreover, the superficial alignment hypothesis Zhou et al. (2024) argues that that almost all knowledge in large language models is learned during pre-training and that alignment tuning only teaches the base LLM in which data format and language style to interact with its users. Further analysis of this hypothesis finds that fine-tuning approaches only influence a small number of tokens focused primarily on stylistic elements Lin et al. (2024). The lack of model expressivity to incorporate constraints in an important challenge for fine-grained constrained text generation.

Format constraints Due to length biases in reward models, LLM models fine-tuned with RLHF tend to suffer from verbosity issues and generate long answers that are not necessarily of high quality Singhal et al. (2023), Park et al. (2024), Dubois et al. (2024b), Kabir et al. (2023), Nakano et al. (2021), Sun et al. (2023), Wu et al. (2024). Offline preference optimization algorithms with implicit reward are known to exploit evaluator length biases and quickly increase the length of the generated text during training without capturing more complex features of human preferences Rafailov et al. (2024), Li et al. (2023a), Dubois et al. (2024a). To disentangle verbosity from quality, length-controlled direct preference alignment methods employ an additional regularization term in the loss function that governs the token length of the generated response Park et al. (2024). Online RLHF approaches that first train a reward model include a similar length regularization term in the reward modeling stage Chen et al. (2024). Among the open problems with current RLHF frameworks is their limited ability to capture and adapt to the complexity of user-dependent preferences in the real world Wang et al. (2024a). Since human preferences can change over time depending on users and their expectations, methods relying on multi-objective reward models are used to capture desirable aspects of the generated texts (eg., verbosity, factuality, helpfulness, harmlessness) Pan et al. (2023), Rame et al. (2024), Dong et al. (2023), Bakker et al. (2022), Wu et al. (2024). To enhance the control and personalization of a single LLM model across desired range of attributes, different user preferences are encoded as unit vectors and embedded numerically into the system prompt Wang et al. (2024a), Yang et al. (2024b), Dong et al. (2023).

5.3 Discriminative approaches

Utility constraints One of the early works to propose constrained generation and manipulation of the generated text learns disentangled latent representations by combining variational auto-encoders with attribute discriminators Hu et al. (2017). Semantic structure is imposed on the latent codes by using global discriminators, one for each attribute, to guide the learning of the discrete text generator and force it to allocate one latent dimension per attribute code. The model is used to generate sentences with constrained sentiment and tense.

Weighted decoding Holtzman et al. (2018) relies on a mixture of discriminative models to guide a recurrent generator towards incorporating attributes that enhance the overall coherence, style, and information content of the generated text. The discriminators complement each other and their weighted contributions form the final decoding objective from the generator. Similarly, stylistic configurations are revised and polished for generated poems by adding additional weights during decoding to control the style of generated poem, including the repetition, alliteration, word length, cursing, sentiment, and concreteness Ghazvininejad et al. (2017). Nevertheless, modifying the scoring function used for generation as in weighted decoding often leads to sacrificing fluency and coherence of the generated text See et al. (2019). Selective sampling Wang et al. (2017) relies on a sample selector (multilayer perceptron for binary classification) which outputs whether the current sample should be accepted or rejected based on the presence of desired target words that define the output style and topic in the generated sequence. The robustness of evaluation metrics is directly correlated with model performance, therefore it is crucial to focus on developing metrics that capture diverse aspects of text quality during training and sampling time.

Generating texts with desirable attributes from a pre-trained unconditional language model $P(X)$ is a non-trivial task. Most approaches resort to either training from scratch a new conditional model $P(X|a)$ for desired attribute a, or fine-tuning $P(X)$ on additional data representative for the attribute a. Theoretically, rejection sampling could also be used to sample $P(X|a)$ from $P(x)$, but this approach is highly inefficient in practice. Fudge Yang & Klein (2021) generates text conditioned on a desired attribute a (for eg., topic control in language generation, degree of formality in machine translation, poetry couplet completion) while only accessing the output probabilities $P(X)$ of generative model G. Given an incomplete sequence prefix, the model trains binary discriminative models for one or multiple desired attributes to predict whether the attribute(s) will be fulfilled in the future complete sequence, therefore evaluation is an important challenge. The output probabilities of the discriminator(s) are then multiplied with the output logits of the generator G to adjust the original probabilities of G accounting for desired attribute(s) a and model $P(X|a)$ via a Bayesian decomposition.

PPLM Dathathri et al. (2019) combines a pre-trained language model with attribute classifiers that guide generation towards specific topics and sentiment styles. These classifiers are trained on top of the last hidden layer of the pre-trained language model, and gradients from the classifiers are backpropagated to update the hidden representations of the language model and steer generation in desirable directions. While PPLM achieves fine-grained control of content and style attributes via a simple gradient-based sampling mechanism, the approach is computationally intensive and inefficient as it requires multiple forward and backward passes for each generation step. Plug-and-play methods have been used to control large pre-trained conversational models such as GPT-2 Radford et al. (2019) using a variety of styles (positive and negative sentiment) and topics (Question, Sport, Business, Finance) Madotto et al. (2020), Liu et al. (2020). Plug-in Language Model (PiLM) Yang et al. (2024a) manipulates the latent state of the language model using a regression model to generate texts that adhere to specific topics or sentiment. Importantly, more effort needs to be focused on collecting datasets for constrained text generation that capture many possible real-world constraints.

GeDi Krause et al. (2020) guides language generation from large language models towards desired attributes by using generative discriminators to compute classification likelihoods for all candidate next tokens on the fly at generation time. Given a class-conditional language model conditioned both on a desired attribute c^+ and an undesired attribute c^-, GeDi-guided contrastive generation uses the two instances of the model as discriminative classifiers to contrast and filter out common attributes between the two classes c^+ and c^-; then aspects of the desired attribute c^+ are transferred across domains via weighted decoding and filtering.

The contrast between a positive and a negative class conditional distribution is employed both at training and inference time to control the bias, toxicity and negativity of GPT-2 Radford et al. (2019) and GPT-3 Brown et al. (2020). Recent approaches explore the role word embeddings can play in debiasing LLMs and steering their generations in particular sentiment directions Subramani et al. (2022), Turner et al. (2023), Li & Liang (2021). LM-Steer Han et al. (2023) leverages the fact that linear transformations in output word embeddings are equivalent to style changes in LLM generation; the method linearly transforms output word embeddings at decoding time using learnt parameters representative for each target style.

5.4 Edit based approaches

Utility constraints Edit based approaches rely on the key idea that changing only a few words or phrases which are indicative of a particular attribute are sufficient to alter the style of a given piece of text. For example, the sentiment of a sentence can be altered from negative to positive by first identifying negative attribute markers ("bad", "worst", "disappointed"), deleting these negative attributes while keeping other content words fixed, and then generating the final output via a recurrent decoder which conditions on the extracted content words and the target attribute Li et al. (2018). Leaving from the observation that humans write text in incremental passes with multiple revisions, a prototype-then-edit model first samples a prototype sentence from the training corpus and then edits it conditioned on an edit vector Guu et al. (2018). Noticeably, text generation based on editing a prototype is much easier compared to generating text from scratch. Also building upon the "Delete Retrieve Generate" framework, the Generative Style Transformer Sudhakar et al. (2019) incorporates a neural mechanism to delete style attributes from the source sentence based on the attention weights of a Transformer model (Delete Transformer), and then generates sentences in the desired target style by decoding with a pre-trained GPT-2 Radford et al. (2018) model.

Activation editing Li et al. (2022a), Hernandez et al. (2023), Li et al. (2024b) methods discover directions in the representation space that correspond to encodings of specific attributes (such as sentiment, topic, style, factual information, truthfulness, etc). When these encodings are added to the internal representations of large language models, they act as knowledge editors by manipulating the generated output to be consistent with desired constraints Kong et al. (2024). The advantage of representation edits is that they enable constrained generation without the need to rely on textual prompts, while enhancing LLM control and intepretability. Steering vectors Subramani et al. (2022), Turner et al. (2023) are added to the hidden states of a language model to generate texts with desired style or sentiment. Other editing approaches manipulate model weights instead of representations by inserting updated factual knowledge directly into model specific parameters Meng et al. (2022a), Meng et al. (2022b), Mitchell et al. (2021), Dai et al. (2022), Rawat et al. (2021), Ilharco et al. (2022), Orgad et al. (2023). Nevertheless, open challenges with these models include indeterminate/uncertain editing boundaries, failing to account for contextual information and entailed consequences of edited facts Liu et al. (2024), Cohen et al. (2024), Zhong et al. (2023). In addition, evaluations focused on assessing the success of model edits mostly consider a few tokens generated after an input prompt, and do not measure the consistency of edits over a long generation of text; failure modes observed in long-form generation include topic drift, lexical cohesion issues, gradual and catastrophic forgetting of previous edited facts - these aspects limit the usefulness of model editing methods at scale Rosati et al. (2024), Gupta et al. (2024), Li et al. (2023c).

Lexical constraints Constrained sentence generation by Metropolis-Hastings sampling Miao et al. (2019) first inserts all constraint keywords in a template in random order, then samples local edit operations (word replacement, deletion or insertion) to perform at specific positions for improving sentence fluency. The probability of each edit operation being accepted or rejected is determined by a language model, however individually sampling each token results in slow convergence. Instead of randomly sampling edit operations, the gradient of a differentiable objective function is used to determine where and how to edit Sha (2020). The majority of editing approaches model a single edit step, unlike humans who do iterative refinement and editing of the content. Modeling the whole process of iteratively generating sequences leverages neural network models to describe the likelihood of multi-step edits, with improved performance over modeling single-order edits Reid & Neubig (2022).

5.5 Adapting existing models and architectures to accommodate constraints

It is non-trivial to impose constraints on existing deep learning models while maintaining high generation quality since their model architecture is designed to generate sentences sequentially from left to right. While current deep learning models are lacking the expressiveness to incorporate constraints at training time and at arbitrary positions in the generated sequence, well known models and architectures are adapted to accommodate constraints through a set of custom engineered approaches. We present these methods below.

Lexical constraints Current architectures used for language generation produce texts sequentially from the first word to the last word, and it is non-trivial to impose lexical constraints on left-to-right generation while maintaining high output quality for natural and fluent texts. Current workarounds for hard lexically constrained text generation address this limitation by generating texts in a non-monotonic fashion when employing ***forward-backward language models***. The backward language model takes a lexical constraint as input, considers it as the starting point and generates the first half of the sentence backwards conditioned on the topic word, while the forward language model takes as input the sequence generated by the backward generator and produces its sentence completion in normal order conditioned on the backward generated sequence. While the topic word can occur at any position in the sentence, this approach can only generate output constrained on at most one lexical constraint; generating sequences with multiple lexical constraints is an open research problem. Moreover, these approaches adapt existing frameworks for constrained text generation by splitting a sentence into two parts, which is unnatural and also hurts fluency when generating half of the sequence in reverse order.

Given a topic word at an arbitrary position in a scientific paper title, a recurrent language model is tasked with generating both past and future words in the title conditioned on the given topic Mou et al. (2015). Similarly, on-topic dialogue responses that satisfy hard lexical constraints are generated with a "sequence to backward and forward sequences" (seq2bf) model Mou et al. (2016) which first predicts a keyword noun that reflects the gist of the response, then decodes the response backward and forward starting from the given word. BFGAN Liu et al. (2019a) employs GANs for lexically constrained text generation using GANs. The model incorporates three modules, namely a backward generator and a forward generator which collaborate on generating lexically constrained sentences, and a discriminator which guides the joint training with policy gradient of the two generators. BFGAN is used to generate Amazon product reviews and conversational responses with lexical constraints.

Generating a fluent sequence which simultaneously satisfies multiple lexical constraints employs a backward-forward LSTM language model to first generate the sequence from a user-defined verb constraint and then satisfy other lexical constraints by word embedding substitution based on cosine similarity between generated tokens and desired constraints Latif et al. (2020). Nevertheless, the approach assumes a verb constraint is always specified in the set of lexical constraints.

Semantic and Utility constraints Steering neural models in specific directions is achieved by: *i)* ***adding special tokens at the beginning or end of the source text***, *ii)* ***incorporating additional conditions into the decoder hidden states*** and *iii)* ***connecting the conditions directly to the decoder output layer***. A topic aware sequence-to-sequence model is used to generate on-topic conversational responses by conditioning the decoder on specific topic words Xing et al. (2016). Imposing conversational goals on dialogue agents aims to guide the conversation towards a designated target subject by combining coarse-grained topic constraints with discourse-level rules Tang et al. (2019). Generating emotional responses in neural conversational systems is achieved by feeding the emotion category embedding to a sequence-to-sequence decoder Zhou et al. (2018). Personalized chit-chat dialogue agents that display consistent personalities, viewpoints and are configurable depending on attributes of the system user are used to produce more personal, specific and engaging dialogue responses Wang et al. (2017), Bosselut et al. (2018), Zhang et al. (2018a). Nevertheless, finding the proper balance between fluency, engagement, consistency and a persistent personality remains an open challenge for current dialogue models due to lack of a measurable objective function and correspondingly suitable evaluation metrics. While we can easily judge whether or not an output satisfies one constraint, it is hard to judge the extent to which (or "how much") it actually satisfies the constraint, and it is even harder to jointly model/measure multiple constraints. Moreover, accounting for repetition and diversity is important as these models often get stuck in an infinite loop of redundant, dull, generic and universally relevant responses that carry little meaning Li et al. (2016c), See et al. (2019), Mou et al. (2016).

For integrating factual knowledge into open-ended conversational systems, factoid and entity-rich web documents are encoded altogether with the conversation history into the same representation which is passed to an attentional neural decoder that generates the response tokens. Similarly, speaker-level representations are integrated into seq2seq conversational models for generating personalized conversation responses Li et al. (2016b). Fact-guided sentence modification for dynamically rewriting, updating or correcting articles according to changing information is an instance of constrained text generation which presents the particular challenge that the rewritten sentence needs to be consistent with an input claim while at the same time preserve non-contradicting content Shah et al. (2020). Given the claim and an old sentence, an updated sentence is produced by first identifying contradictory components in the input sentence, masking these, then using the residual sentence and the claim as input into a two encoder sequence-to-sequence model with copy attention to produce the update sentence consistent with the claim. Syntactically controlled paraphrase generation produces paraphrases of an input sentence by constraining the system on the target syntactic form Iyyer et al. (2018), however not many syntactically constrained datasets to learn from are available.

Controllable story generation based on RNNs is used to influence the story ending valence (whether happy or sad) and the storyline (specified as a sequence of words) Peng et al. (2018). Story-telling methods commonly use a hierarchical approach to thematically consistent story generation, by first generating a prompt describing the topic for the story, and then constraining on the prompt for generating the story content Fan et al. (2018b); additionally, constraints on the presence of entities are included as well Clark et al. (2018). Open-domain story generation requires composing coherent natural language texts that describe plausible sequence of events and is more challenging compared to generating stories in a narrow domain given an existing plot.

Unsupervised machine translation methods are adapted for the task of text-style transfer by incorporating stylistic constraints in a neural seq2seq model with attention and ***using a style classifier to guarantee the accuracy of style transfer*** Zhang et al. (2018c), or for control over multiple style attributes, including gender, sentiment or product type Lample et al. (2018). In machine translation, honorifics constraints are important for producing socially appropriate forms of address and controling the level of courtesy Sennrich et al. (2016); the system user defines the desired level of politeness of the translation, however these user-defined constraints are only soft constraints and can be overridden by the attentional encoder-decoder machine translation system whenever the source text provides strong politeness clues.

For effective imposition of semantic structure in constrained text generation, latent space representations need to be disentangled John et al. (2019), such that varying an individual latent code will only change a single desired attribute. VAEs can achieve meaningful latent representations with designated semantics when combined with ***attribute discriminators*** and optimized end-to-end with differentiable softmax approximation Hu et al. (2017); this allows to generate sentences with constraints on sentiment and tense. Given an input sequence and a set of labels, sequence transduction with multi-space variational autoencoders Zhou & Neubig (2017) generates an output sequence that alters the content of the input sequence according to the constraints specified by the labels; the method is used for morphological inflection in multiple languages. In general, constrained text generation approaches assume that constraints need to be known a priori; however, this is not always possible, for eg., when suggesting alternative phrases for search queries in real-time, or when generating responses in dialogue systems according to the dynamics of the conversational context. Recent constrained text generation approaches control attributes of a generated sequence based on another sentence example: given two sentences X and Y, the goal is to generate a new sentence Z that follows the semantics of X and the syntax of Y. To this end, a VAE model with two latent variables is used to achieve disentanglement in the continuous latent space between syntax and semantics Chen et al. (2019), Bao et al. (2019). Topic guided VAEs Wang et al. (2019) use a Gaussian mixture model prior where each mixture component corresponds to a latent topic extracted from data as opposed to using pre-defined parameter settings which do not incorporate semantic meaning into the latent codes; the model is used for text summarization with designated topic guidance. Abstractive and extractive sentence compression with VAEs assumes the existence of a background language model from which a latent summary sentence is drawn first, and then the observed sentence is generated conditioned on the latent summary Miao & Blunsom (2016); the model is able to balance copying a word from the source sentence with generating it from the background distribution. Iterative refinement of a sequence to transform it into another sequence with desired attributes exploits geometry of the latent space to produce incremental higher-quality revisions with theoretical guarantees in the combinatorial space of sequence elements Mueller et al. (2017), Shen et al. (2017). Such latent variable manipulations allow to rewrite modern text in the language of Shakespeare, improve sentence positivity, address word substitution and word order recovery tasks without need for any revision examples. Constraints on the use of metaphor and personification in poems are incorporated in a conditional VAE with a rhetorically controlled decoder trained to emit meaningful and diverse rhetoric and overcome generic sentences Liu et al. (2019b). Variational neural machine translation Zhang et al. (2016) incorporates a continuous latent variable to model the underlying semantics of sentence pairs. Nevertheless, efficiently performing posterior inference and large-scale training during the incorporation of latent variables remains an open challenge for constrained VAEs. Finding structure in the latent space that corresponds to particular sentiment makes it possible to steer the generation towards desired sentiment by adding the sentiment direction vector to the residual stream when generating sentence completions Tigges et al. (2023).

Modifying textual attributes of sentences including sentiment, style, tense, voice, mood and negation is achieved by ***incorporating conditioning information into a neural encoder-decoder model***, and optimizing a reconstruction loss which interpolates between auto-encoding and back-translation components to encourage content compatibility, as well as an adversarial loss which encourages sentence-level stylistic attribute compatibility Logeswaran et al. (2018). The model allows simultaneous conditioning on multiple textual attributes, however the extent to which the generated sentences match the conditioning information requires new objective evaluation metrics for attribute accuracy and content compatibility/preservation.

Style transfer between scientific papers and newspapers is performed with ***separate style decoders***, or by generating both content and style from the same decoder Fu et al. (2018). In poetry generation, it is common to impose hard constraints on rhyme, rhythm, and topic Ghazvininejad et al. (2016), Ghazvininejad et al. (2017). Given a user-supplied topic, the poetry generation algorithm first generates a large set of on-topic words and phrases, assigns rhyming words and phrases to specific lines, and then combines finite-state machinery with an RNN language model to score plausible poems that meet the desired constraints. While ***augmenting an RNN with a working memory*** to explicitly maintain a limited history of generated topics and context, coherence in meaning and topics across the overall poem remains an important challenge Zhang & Lapata (2014). Constrained recurrent models are also used to generate online product reviews of certain topic, sentiment, style and length Ficler & Goldberg (2017), affective dialogue responses Ghosh et al. (2017), or for modeling participant roles and topics in conversational systems Mei et al. (2017).

Alternative non-autoregressive architectures based on continuous diffusion models are adapted for text generation with semantic and syntactic constraints Yang et al. (2022), Austin et al. (2021), Gong et al. (2022). Diffusion-LM Li et al. (2022b) gradually denoises a sequence of Gaussian noise vectors into word vectors, resulting in a hierarchy of continuous latent representations which enables gradient-based methods to steer the text generation process. Nevertheless, training of diffusion models is slower to converge and decoding from these models takes longer time. To speed up the inference process, adaptive sampling strategies are applied for different generation stages in the context of story generation Tang et al. (2023).

Format and Utility constraints Text simplification models parameterized on constraints such as length, amount of paraphrasing, degree of lexical and syntactic complexity are used for generating texts easier to read and understand with simpler grammar and structure Martin et al. (2020). Towards a similar goal of controlling the degree of lexical complexity, the ***training loss function is changed to assign weights to words based on their complexity level*** Nishihara et al. (2019). In text summarization, constraints on the output sequence length for neural encoder-decoder models are specified as ***length embeddings*** and are passed as additional input to the decoder Kikuchi et al. (2016).

Faithfulness in abstractive text summarization is enforced in a seq2seq model by conditioning on both the source text and extracted factual descriptions Cao et al. (2018); this helps avoid generating false facts in the output summary. Hybrid text summarization approaches combine an unsupervised sentence extractor which selects salient sentences from the input document with a sentence abstractor that paraphrases each extracted sentence to overcome limitations of parallel aligned datasets Nikolov & Hahnloser (2020).

Reinforcement learning is used in the context of constrained natural language generation to directly optimize non-differentiable reward functions and evaluation metrics. While any user-defined reward function can be employed for training, most frequently optimized metrics with RL are BLEU for machine translation Ranzato et al. (2015), ROUGE for text summarization Ranzato et al. (2015), Paulus et al. (2017), Wu & Hu (2018), Gao et al. (2019b), or human-defined conversation metrics focused on coherence, informativeness, sentiment, politeness, toxicity, question, repetition or semantic similarity Li et al. (2016c), Saleh et al. (2019), Wu & Hu (2018). However, manually defined reward functions based on heuristics cannot cover all crucial aspects of a natural realistic conversation Bosselut et al. (2018), Gao et al. (2019b). In addition, rewards are commonly modeled at the word level accounting for the probability of generating each word in a sentence Ranzato et al. (2015), Jaques et al. (2019); such low-level control makes credit assignment challenging since the number of actions available to the RL agent is equivalent to the number of words in the vocabulary. Defining a global score that measures complex aspects of text quality beyond local n-gram patterns and which can reliably approximate human judgments of text quality remains an open challenge Bosselut et al. (2018).

In the RL framework the generative model is seen as an agent with parameters that define a policy and which interacts with an external environment by taking actions, receives a reward once it reaches the end of a sequence and updates its internal state consequently. To this end, policy gradient methods are used to train text generative models and alleviate issues such as exposure bias and loss functions which do not operate at the sequence level. However, policy gradient algorithms present large variance and generally struggle in settings with large action spaces such as natural language generation. In addition, they take very long time to converge Choshen et al. (2019) and the improvement in the optimized metrics is not always reflected in human evaluations of text quality. Training RL models to optimize n-gram evaluation measures based on local patterns provides only a limited and myopic perspective of overall text quality and does not necessarily lead to better text quality, overall coherence or discourse structure Bosselut et al. (2018). Moreover, fine-tuning on such measures may yield deteriorated outputs despite increased automatic scores, while difficulty in constrained optimization with RL often leads to sparse, non-informative and delayed reward signals.

Learning RL rewards from human preferences aims to incorporate human feedback in text generation and teach models to follow human instructions Ouyang et al. (2022), OpenAI (2022), OpenAI (2023). Neural reward learning schemes train neural teachers that learn to score an ordered sequence of sentences and formulate rewards that guide coherent long text generation Bosselut et al. (2018); the approach is used for generating cooking recipes given the dish title and the set of ingredients as constraints. Learning-to-rank algorithms are used to approximate ground-truth oracle rewards in extractive multi-document summarization to indicate the quality of a summary or preferences over summary pairs Gao et al. (2019b). Machine learnability of human rewards in neural machine translation models is approached by first training reward estimators on rewards collected from offline logs, then integrating these reward estimators in an off-policy RL setting Kreutzer et al. (2018). Similarly, implicit human reactions such as sentiment or length of a conversation are used to learn rewards for fine-tuning off-policy RL models for dialog Jaques et al. (2019). Nevertheless, human feedback is noisy, not well-defined, complex and inconsistent. Using RL to improve system outputs with respect to human-centered metrics of conversation quality is highly dependent on developing robust metrics tailored to the particular application domain, for eg. increasing politeness of a technical-support system or reducing toxicity of generated language.

Hard-constrained text generation in a non-monotonic order relies on a tree-based text generation scheme, where a word is generated at an arbitrary position in the sentence, then binary trees of words to its left and right are recursively generated Welleck et al. (2019). Learning proceeds in an incremental fashion in an imitation learning framework, where the policy gradually moves from imitating the oracle to reinforcing its own preferences and generating texts without a pre-specified word order. Nevertheless, the time complexity of the approach is $\mathcal{O}(n)$, same as for autoregressive models and the constructed tree does not reflect a high-level to low-level hierarchy of concepts.

5.6 Prompting Large Language Models

The paradigm of prompt-based learning, which became popular with the release of OpenAI GPT-3 Brown et al. (2020) model, demonstrates it is possible to elicit factual and commonsense knowledge from large language models and steer them towards desired behaviours via a textual prompt. Instead of adapting models to downstream tasks via objective engineering as it is common during fine-tuning, prompt-based learning reformulates downstream tasks to resemble those encountered during the language model pre-training phase where a fill-in-the-blanks objective is used Liu et al. (2023a). While prompting allows for manipulating the model behaviour to predict desired output, sometimes even without additional task-specific training, model performance on a given task is highly dependent on the quality of the prompt used to steer the model and how much conditioning text can fit into the model's input. In general, identifying the most appropriate prompt for a task is a challenge in itself. While prompting provides a natural interface for humans to communicate with machines, human users have little knowledge of which instructions are compatible with a given model and need to experiment with a wide range of discrete prompts to find suitable ones that elicit desired behaviours Zhou et al. (2022). Given that plain language prompts do not always produce the intended results, automated methods for prompt design are proposed in the literature, including searching over the discrete space of words guided by training data Shin et al. (2020), prefix tuning which optimizes a task-specific continuous vector Li & Liang (2021), Hambardzumyan et al. (2021), prompt tuning which learns soft prompts via backpropagation Lester et al. (2021), Qin & Eisner (2021), natural language prompt engineering where large language models themselves generate meta-prompts for solving a wide range of tasks Reynolds & McDonell (2021), Zhou et al. (2022) or inverse prompting which uses the generated text to inversely predict the prompt Zou et al. (2021). Directional Stimulus Prompting Li et al. (2023b) guides black-box language models such as ChatGPT OpenAI (2022) towards desired outputs by optimizing a policy model trained to maximize rewards that measure the alignment between the generated text and desired topics and keywords on tasks such as text summarization and dialogue response generation. Steering LLMs to generate texts that reflect multiple perspectives and diverse opinions is achieved by first modeling data-driven personas when embedding individuals and their viewpoints into a continuous vector space, then using soft prompting techniques to map persona embeddings to specific tokens Li et al. (2024a), Hwang et al. (2023), Santurkar et al. (2023).

While there is not much theoretical understanding behind the reasons why and how prompting works, it is assumed that prompting provides a way to steer large language models in particular directions by helping locate a specific task in the pre-trained model's existing space of learned tasks, phenomenon evidenced by the superior performance of some prompts over others Reynolds & McDonell (2021). Nevertheless, prompting large language models is far from sufficient for robust and reliable constrained text generation. Prompting approaches must be employed with caution, as models can deviate from the original prompt, fail to maintain the coherence and produce texts on unrelated topics Zou et al. (2021), Hernandez et al. (2023), and may even degenerate into toxic text from seemingly innocuous prompts Gehman et al. (2020). In improving prompting reliability, it is important to account for generalization outside of distribution, reducing social biases, ensuring fairness to different demographic groups, calibrating output probabilities and updating the model's factual knowledge and reasoning chains Si et al. (2022).

Prompting Considerations Large language models leverage vast amounts of information they learn from web-scale pre-training datasets which they store in their parameters, resulting in improved performance on many knowledge-intensive tasks Brown et al. (2020), OpenAI (2022), OpenAI (2023). Nevertheless, it is important to understand what kind of knowledge LLMs actually capture. In factuality assessments of LLMs, it is found that current systems tend to hallucinate and make up facts Maynez et al. (2020), Tam et al. (2022), Zhou et al. (2021), Lin et al. (2022), and this behaviour becomes more predominant as the rarity of entities increases Min et al. (2023). There is a strong correlation between the correctness of answering factoid questions and the number of pre-training documents relevant to that question Kandpal et al. (2023); models are more accurate on instances whose terms are more prevalent in the training data, and struggle on questions containing long-tail terms with low document count. Similarly, mathematical reasoning capabilities are correlated with training data frequency, and the selection of the training corpus does impact the few-shot performance of LLMs Razeghi et al. (2022), Shin et al. (2022). These findings suggest that low-order co-occurrence statistics in the pre-training dataset have a significant impact on model performance, leaving the open question of the extent to which current models generalize beyond the training data. Ideally, a general purpose language model is able to generalize not only to unseen instances of known tasks but to new tasks as well, however current LLMs are found to rely on narrow, non-transferable procedures for task solving specialized to tasks seen during pre-training Wu et al. (2023); in counterfactual settings their performance degrades considerably, indicating overfitting to training tasks.

6 Constrained NLG Evaluation

Evaluation of constrained text generation is performed using the same evaluation approaches and methodologies available in the natural language generation literature. In general, evaluation of the generated text is largely an unsolved and notoriously difficult problem Borji (2019). Currently, there is no well-established consensus on how NLG systems should be evaluated, van der Lee et al. (2019), Gkatzia & Mahamood (2015), and the lack of meaningful quantitative evaluation metrics to accurately assess the quality of trained models is detrimental to the progress of the field. In the absence of well established evaluation measures, natural language evaluations are carried in a rather ad-hoc manner with a lot of variability across the proposed models and tasks on inconsistent benchmarks, resulting in misleading performance measures. Subjective evaluations based on visual inspection of the generated samples often lack scientific rigour, making it difficult to quantify and judge precisely the quality of a generative model Hashimoto et al. (2019). In what follows we review the main methods for constrained text generation evaluation.

Lexical constraints Measuring how many of the given lexical constraints are included in the generated outputs is done using ***concept coverage*** Lin et al. (2019), Lu et al. (2021); the metric is computed as the the average percentage of input concepts that are present in the lemmatized outputs.

Semantic and syntactic constraints Surface similarity based on ***n-gram overlap metrics***, such as BLEU Papineni et al. (2002), ROUGE Lin (2004), METEOR Banerjee & Lavie (2005) measure to what extent the generative model can preserve content by retaining words commonly shared between the generated output and ground-truth references. Such metrics are commonly used to measure response relevance in dialogue systems Galley et al. (2015), Li et al. (2016b), translation quality in neural machine translation Sennrich et al. (2016), assess summary quality in text summarization See et al. (2017). In general, the correlation between word overlap metrics and true text quality is a widely debated topic Li et al. (2016c). Evaluation metrics based on local n-gram patterns only provide a limited and myopic perspective of overall text quality and are notoriously poor at evaluating dialogue systems Liu et al. (2016), See et al. (2019), Bosselut et al. (2018).

Perplexity Jelinek et al. (1977) based evaluation metrics are used to evaluate and compare language models, and measure the fluency and diversity of the generated samples Madotto et al. (2020), Bosselut et al. (2018), Li et al. (2016b). Reverse Perplexity Zhao et al. (2018) and Forward Perplexity Kim et al. (2017) scores are calculated by training language models on synthetic samples, respectively real samples, and then using these trained models to measure perplexity real samples, respectively generated samples. Nevertheless, perplexity is a model dependent metric, and "how likely a sentence is generated by a given model" is not directly comparable across different models. Moreover, numerous studies find perplexity to be an inadequate measure of text quality Theis et al. (2016), Fedus et al. (2018), since models with high likelihood can generate low-quality samples, while samples of good quality can present low likelihood. In addition, infinite perplexity can still be obtained from a perfect model even when its ability to generate test sentences is removed Hashimoto et al. (2019).

P, R, F1 are used to measure the distance of the generated samples to the real data manifold Lucic et al. (2018). When precision is high, the generated samples are close to the data manifold, and when recall is high, the generator outputs samples that cover the manifold well. Metrics that aggregate precision and recall such as F_β, a generalization of the F_1 score, are used to quantify the relative importance of precision and recall Sajjadi et al. (2018). Nevertheless, the data manifold of non-synthetic data is unknown and therefore impossible to compute in practice.

Content diversity measures how different the generated sentences are from each other, by either considering word choice, topic and meaning Vijayakumar et al. (2016), Gimpel et al. (2013), Ippolito et al. (2018), or by looking at the level of sentence interestingness or unlikeliness Hashimoto et al. (2019). Perplexity on a reference set, n-gram diversity Li et al. (2016a) and Self-BLEU Zhu et al. (2018) are commonly used measures of the diversity of the generated samples. In addition, Backward-BLEU Shi et al. (2018) evaluates test data using the generated samples as reference; the higher the score the more diverse the generator output. Lexical diversity Bache et al. (2013) calculates the ratio of unique tokens to the total number of generated tokens.

Similarly, Distinct-k or Dist-k Li et al. (2016a) measures the total number of unique k-grams normalized by the total number of generated k-gram tokens to avoid favoring long sentences. Nevertheless, the Dist-k metric ignores the fact that infrequent k-grams contribute more to diversity than frequent ones and assign same weight to all k-grams that appear at least once. Distinct-1 and Distinct-2 are used to measure the diversity of constrained conversational responses Baheti et al. (2018), Zhang et al. (2018b) and rhetoric constrained generated poems Liu et al. (2019b). Entropy based metrics such as Ent-k Zhang et al. (2018b) reflect the frequency difference of k-grams and to analyze the information content of the generated responses in dialogue systems Serban et al. (2017), Mou et al. (2016).

Unlike traditional evaluation metrics based on heuristics, learnable metrics train machine learning models on human annotated datasets to learn a scoring function that reproduces human judgements. ***Fully-learnt metrics*** leverage existing datasets of human ratings to learn automated evaluation metrics that fit the human data distribution, and can be tuned to measure specific properties of the generated texts, such as fluency, style, grammaticality, fidelity, etc. Linear regression based on human judgements is used to learn a model for scoring system summaries Peyrard et al. (2017). RUSE Shimanaka et al. (2018) combines sentence embeddings in a multi-layer perceptron regressor model. ESIM Chen et al. (2017), Mathur et al. (2019) feeds the encoded representations of the candidate and the reference sentence into a feedforward regressor. BLEURT Sellam et al. (2020) fine-tunes BERT Devlin et al. (2018) on human ratings datasets for similarity score prediction. MAUDE Sinha et al. (2020) is proposed for the evaluation of online dialogue conversations and leverages sentence representations from pre-trained BERT to train text encoders which can distinguish between valid dialogue responses and fake examples. BARTScore Yuan et al. (2021) formulates the evaluation of generated text as a text generation task from pre-trained language models and measures the weighted probability of the generated text given another text as input or output. GPT Judge Lin et al. (2022) fine-tunes GPT3 Brown et al. (2020) model on human annotated data to clasify answers of QA systems as true or false, evaluating factuality and truthfulness. The same evaluation metric, this time based on GPT-4 OpenAI (2023), is used to establish via prompting whether texts generated by GPT-4 are more similar to human-written reference answers or GPT-3 machine-generated texts. Nevertheless, the GPT-4 evaluator is known to be biased in its preferences towards answers with longer length Singhal et al. (2023), Wang et al. (2023b), Li et al. (2023a). GPTScore Fu et al. (2023) computes the conditional probability of generating the target text given specific context. FactScore Min et al. (2023) breaks generation into atomic pieces of information and evaluates the factual precision of long-form text by measuring the percentage of atomic facts supported by a reliable knowledge source. Other evaluation metrics based on probabilities inferred from pre-trained masked language models include InfoLM Colombo et al. (2022), CTRLEval Ke et al. (2022), MaskEval Liu et al. (2022). ***Hybrid metrics*** combine learnt elements with human-defined logical rules, for example, contextual embeddings with token alignment rules. BERTscore Zhang et al. (2019) evaluates generated text against gold standard references using soft-string similarity matches (i.e. cosine similarity) computed on pre-trained contextualized BERT Devlin et al. (2018) token embeddings. MoverScore Zhao et al. (2019) combines contextualized representations of system and reference texts with semantic measures of distance computed using Word Mover's Distance Kusner et al. (2015); the metric is extended to evaluate multi-sentence texts Clark et al. (2019). Human and statistical evaluation are combined in HUSE Hashimoto et al. (2019), an evaluation framework which estimates the optimal error rate of predicting whether a piece of text is human-written or machine-generated. However, a limitation of learned evaluation metrics is that they generally fail to generalize well across different systems Chaganty et al. (2018).

Utility constraints A commonly used approach in the literature to assess whether generated texts have desirable attributes is to rely on an attribute classifier and measure the ***classification score***, i.e. the fraction of outputs generated by the model having the desired attribute Hu et al. (2017), Shen et al. (2017), Li et al. (2018). ***Adversarial evaluation*** Bowman et al. (2016), Kannan & Vinyals (2017) employs an evaluator trained to distinguish machine-generated from human-written texts, analogous to the discriminator in GANs Goodfellow et al. (2014). On this note, ***pre-trained attribute classifiers*** and ***class-specific discriminators*** measure how well the generated samples match the conditioning labels on attributes such as sentiment, tense, voice, mood and negation Logeswaran et al. (2018), Li et al. (2017), Bruni & Fernández (2017), guarantee the accuracy of stylistic text transfer Zhang et al. (2018c), Shen et al. (2017), or are used to evaluate biases against certain demographics and quantify model fairness in downstream settings Cao et al.

(2022), Mathew et al. (2021), Kurita et al. (2019). ***GLEU*** Napoles et al. (2015) was originally proposed for grammatical error correction, and later adopted for the evaluation of text style transfer since both tasks require localized edits to the input sentence; GLEU is found to present a reasonable balance between target style match and content retention Sudhakar et al. (2019).

Readability metrics such as Flesch-Kincaid Grade Level Kincaid et al. (1975) and Flesch Reading Ease Flesch (1979) are used to account for simplicity and measure the reading difficulty of a piece of text. Both metrics are computed as linear combinations of the number of words per sentence and number of syllables per word with different weighting factors. Even though these metrics are frequently used to measure readability, are fast and easy to compute, they should not be used on their own but in combination with metrics that capture the grammaticality and meaning preservation of the generated output Wubben et al. (2012). In addition, they were not designed for measuring text readability in scientific or specialized domains, and are only available for the English language.

All constraints While automated evaluation helps assess generated texts quickly and cheaply, the use of automated evaluation metrics is dependent upon their correlation with human judgements of quality Fomicheva & Specia (2019). ***Human evaluations*** remain the gold-standard in natural language generation and automated evaluation metrics can only be used as a proxy for human judgements only when there is reasonable correlation with human decisions. Ideally, automated evaluations are carried simultaneously with human annotation studies, and not as a replacement of human evaluations. In text style transfer, human evaluations are conducted to determine how accurately constrained text generation methods identify stylistic textual attributes in the source input and replace these with desired target attributes in generated sentences Sudhakar et al. (2019). In conversational systems, responses generated by open-domain chatbots are evaluated across two dimensions: *i)* humanness, as a proxy for the fluency and coherence of the generated responses, and *ii)* attribute consistency, to determine whether the style and topic enforced by the generation model are well captured Madotto et al. (2020). Human evaluations are also carried to determine the plausability of the generated response, as well as to measure its content richness and how much new information it adds to the conversation) Baheti et al. (2018). Outputs generated by neural conversational systems are also assessed for quality, style and topic to determine whether the acquisition of styles of famous personalities, characters, or professionals is achievable, and whether the conversational topic can be steered in particular directions Wang et al. (2017).

Limitations of current evaluation metrics Given the wide diversity of evaluation paradigms in the field of NLG, it becomes challenging to objectively compare models and research progress when different evaluation metrics are employed in each work. By far, human-quality texts are considered the ground-truth for evaluating the output of NLG systems, serving as an upper bound measure of their performance. However, collecting human-quality texts and/or soliciting human judgements of text quality is a costly and time-consuming process which requires careful design choices. Often times automated metrics that present reasonable correlation with human evaluations of text quality are used as a proxy for human judgements, however these metrics come with their own limitations. A common complaint is the lack of good ways to encode what constitutes human-quality output in an automated metric Clark et al. (2021). In addition, shortcomings of current evaluation metrics include poor correlations with human judgements, lack of interpretability of their scores, the presence of complex biases in their evaluations Dubois et al. (2024a), poor adaptability across tasks and inability to capture all nuances in a task Khapra & Sai (2021). In what follows we discuss limitations of existing evaluation metrics hoping to inform on the development of more robust evaluations for NLG systems.

Word Overlap metrics measure the lexical overlap between the model generated text and a set of human-written references. Metrics such as BLEU Papineni et al. (2002), ROUGE Lin (2004) and METEOR Banerjee & Lavie (2005) allow for fast and inexpensive development cycles and have been widely adopted for evaluating the output of natural language generation systems based on their correlation with human judgements at the time they were introduced, nevertheless their use is not without problems. On the one hand, the choice and quality of references is critical for improving the correlation between human and automated evaluation Freitag et al. (2020). Current evaluation metrics are biased towards assigning higher scores to outputs that share a similar style with the reference, therefore collecting only a single style of references fails to reward systems that produce alternative but equally accurate outputs Popović (2019); besides, collecting human-written references for new tasks is costly. On the other hand, these metrics assume that valid machine-generated responses present a significant degree of overlap with ground-truth references; this is problematic for open-ended text generation tasks that require diversity and creativity (for eg., dialogue generation), and in such cases their correlation with human judgements is relatively low Liu et al. (2023c), Liu et al. (2016), Graham et al. (2019), Sellam et al. (2020). For the evaluation of text simplification, BLEU presents weak or no correlation with grammaticality and meaning preservation for sentence splitting operations, therefore penalizing simpler sentences Sulem et al. (2018). In addition, improvements in BLEU do not necessarily reflect an improvement in machine translation quality and there is a huge amount of variation for identically scored hypotheses Callison-Burch et al. (2006) (i.e. a wide variety of candidate outputs receive the same score when they present the same degree of overlap with the reference although they greatly vary). Word overlap metrics are also insufficient for measuring factual correctness of text summarization and fail to correlate with human judgements of factuality Falke et al. (2019), Kryscinski et al. (2019), Pagnoni et al. (2021). Even more concerning is that the great majority of automated metrics, and in particular conventional reference-based metrics such as BLEU Papineni et al. (2002) and CIDER Vedantam et al. (2015), are found to overrate machine-generated text over human-written text even though the machine text falls short of humans Kasai et al. (2022). In addition, BLEU and ROUGE fail to accurately measure content quality, capture syntactic errors and do not reflect the reliability of NLG systems Reiter & Belz (2009), Stent et al. (2005). Using such evaluation metrics to compare systems may lead to drawing inaccurate conclusions, gives the false impression of progress to the research community and actively discourages the development of stronger human-like generative models.

Since BLEU is based on n-gram precision, lexical differences between the hypothesis and references are aggressively penalized even when they are similar or synonymous to the reference. Given that no partial credit is given if an n-gram does not exactly match a sub-sequence of the reference, BLEU is also hard to optimize due to the fact that learning objective is flat and cannot hill-climb through intermediate hypotheses that have high semantic similarity or synonymy, but low n-gram overlap Wieting et al. (2019). Alternative metrics based on word embeddings are easier to optimize as they output continuous values and capture fine-grained distinctions between similar outputs Wieting et al. (2019). When used for measuring the quality of back-translations for data augmentation, BLEU only shows significant improvements for test examples if the source itself is a translation Edunov et al. (2020); whenever references are translations and the source itself is natural text, BLEU fails to capture human preference for source original sentences. While the use of multiple references substantially improves reference-based metrics, often times evaluations are conducted using a single human-written reference per instance; in such cases strong referenceless metrics frequently achieve higher correlation with human judgements Rei et al. (2020). Developing evaluation metrics that correlate well with human judgements on an instance level could serve to augment and validate human annotations.

To overcome the limitations of reference-based evaluation metrics, reference-free natural language evaluators are proposed Fu et al. (2023), Wang et al. (2023a). Simultaneously, evaluating the quality of generated texts based on a form-filling paradigm leverages large language models with chain-of-thoughts Liu et al. (2023c): given a prompt that defines the evaluation task and desired evaluation criteria, the language model generates a chain-of-thought with detailed evaluation instructions based on which it will then score the generated text according to the defined criteria. While LLM-based metrics seem to outperform reference-based and reference-free evaluation metrics in terms of correlation with human judgements for open-ended and creative NLG tasks, they are very sensitive to the instructions and prompts given. Moreover, they tend to prefer LLM-generated texts over high quality human-written texts, which leads to biased predictions especially when used as reward signal for improving themselves. Using language models for "self-evaluation" indicates their predictions are well calibrated for token probabilites in-distribution, but they struggle with calibration in settings outside of the data distribution Kadavath et al. (2022).

Model-Based Evaluation metrics are becoming increasingly popular for NLG evaluation due to powerful representations learnt by pre-trained language models and high correlations with human judgements of text quality. However, current language models have well-known flaws and limitations, for example they assign high likelihood to degenerate texts, i.e. output that is bland, incoherent, or repetitive Holtzman et al. (2019), can be insensitive to perturbations such as word order randomization Pham et al. (2021), negation Ettinger (2020) or named entity replacements Balasubramanian et al. (2020), exploit superficial cues through the use of the self-attention mechanism Pham et al. (2021) and exhibit naive understanding of the meaning of sentences without complex reasoning Ribeiro et al. (2020). To investigate the extent to which model-based evaluation metrics suffer from the same limitations as black-box pre-trained language models, stress tests are used to complement human correlation tests and detect the blind spots of evaluation metrics He et al. (2022). The authors construct a noised hypothesis set by applying different synthetic errors to ground-truth human-written references; if this noised hypothesis set is not scored worse than the original unperturbed set, it means the evaluation metric fails the corresponding stress test. Stress tests reveal that model-based evaluation metrics can be insensitive to errors at the start and middle of the generations when based on pre-trained models that do not encode long-range context, their judgement can be mislead by simply injecting valueless text spans into the hypotheses, are biased towards frequent n-grams, present a self-evaluation bias by unfairly ranking generations from their underlying base pre-trained language model higher than better quality generations from larger models, fail fluency tests (lemmatizing verbs, removing articles, prepositions or tokens at the end of the hypothesis) and consistency tests (sentence switching, replacement or negation). Complex biases, including a strong preference for longer outputs, are a common issue when relying on LLMs models to estimate response quality Li et al. (2023a), Dubois et al. (2024a). While some model-based metrics perform better than others, it is important to recognize their limitations and use each metric with awareness of its blind spots. To mitigate the risks of drawing inaccurate conclusions based on a single metric, using combinations of evaluation metrics that cover each other's blind spots is recommended. For example, evaluation metrics based on pre-trained language models that encode long range context could be more robust to errors in the beginning or middle of the generations, valueless text span injections can be identified by word-overlap based metrics such as ROUGE Lin (2004), biases towards frequent n-grams can be detected by using diversity metrics, and truncation errors can be recognized via precision, recall and F1 scores. In addition, to mitigate unfair biases it is desirable to avoid using the same pre-trained language model for generation as well as base for the evaluation metric, or comparing different pre-trained models using an evaluation metric that relies on one of these models. Finally, adding explainability on top of black-box evaluation metrics can help identify system quality issues and increase trust in the evaluation of NLG systems Leiter et al. (2022).

Human Evaluation is considered the gold standard for the evaluation of NLG systems, however there is no consensus on how these human studies should be conducted Gkatzia & Mahamood (2015), van der Lee et al. (2019). The large variability in the design of human evaluations leads to difficulty in comparing results across different studies and also impacts the reliability of the inferred conclusions. The lack of consistency in human evaluation can be attributed to different factors such as the level of expertise of human annotators, their cognitive biases, ambiguity of the annotation task itself, or the actual wording of questions and instructions presented to participants, i.e. “how something is asked as opposed to what is asked” Schoch et al. (2020). Untrained human evaluators may provide inconsistent results and contradictory reasons behind their judgments, leading researchers to state that "all that's human is not gold" Clark et al. (2021). Unsurprisingly, selecting a different subset of annotators can lead to different conclusions due to variations in individual annotators' understanding of the annotation scheme Amidei et al. (2020); in general, it is hard to decompose, interpret and validate crowdworker evaluations Kasai et al. (2022). Depending on the evaluation setup, it may be sensible to use qualified evaluators who have gone through an extended training and can provide more reliable annotations. Moreover, improving the robustness and transparency of human evaluation guidelines is essential for increasing the reliability of human annotations. As the fluency of generated texts is improving, it is important to not only focus on surface-level aspects of text quality when conducting human evaluations, but to also assess the informativeness and usefulness of generated texts in downstream settings Clark et al. (2021).

Future Outlook While so far we have reviewed limitations of existing evaluation metrics, we would also like to note the metrics that are missing or are under-represented in the literature, particularly metrics for measuring the trustworthiness, factuality, fairness, bias, toxicity, efficiency, diversity, uncertainty quantification, calibration and robustness of text information systems. In addition, it is important for the community to focus on the interpretability aspect of evaluation metrics, particularly for model-based evaluations that currently function as a black-box Leiter et al. (2022). In the era of large language models, aspects such as knowledge, reasoning, memorization/copyright and disinformation are becoming increasingly important to quantify and analyze for NLG systems Liang et al. (2022). Special attention also needs to be paid to existing datasets used to evaluate the generalization abilities of state-of-the-art methods to ensure there is no overlap between the train set and the test set; in such cases, evaluations inadvertently measure memorization instead of the model's ability to generalize, giving the false impression of improvements in performance Elangovan et al. (2021). Large language models in particular are known to memorize parts of their training data, phenomenon which becomes more predominant with increasing the model capacity and the repetition of training examples Carlini et al. (2022), Razeghi et al. (2022). On top of this, given that LLMs are trained on web-scale datasets, evaluations are subject to potential data contamination issues Wu et al. (2023), Dodge et al. (2021), Magar & Schwartz (2022). Therefore, interpreting evaluation results must be done with caution accounting for the source pre-training data in determining to what extent current models generalize vs simply memorize training examples Razeghi et al. (2022); this also highlights the need to reconsider and redefine evaluation schemes for LLMs, and focus on debiasing current evaluation metrics Li et al. (2023a), Dubois et al. (2024a). Finally, it is important to consider how advances in generative models can benefit and inform the development of more suitable evaluation techniques, and vice versa. Bidimensional leaderboards Kasai et al. (2022) that simultaneously track progress in language generation models and evaluation metrics can bridge the gap between generation modeling and evaluation research. As generation models continue to improve, it is important to keep reassessing and updating evaluation metrics so that they accurately reflect the target objectives and correlate with human language use in the real world Zellers et al. (2021).

7 Constrained NLG Benchmarks and Datasets

The collection of datasets that capture a wide diversity of constraints and are representative of many real world situations are critical for advancing safe and robust constrained text generation. Existing benchmarks focused on politeness Madaan et al. (2020), formality Rao & Tetreault (2018), sentiment Shen et al. (2017), writing style Jhamtani et al. (2017) are rather limited in nature and do not offer fine-grained control over stylistic attributes. StylePTB Lyu et al. (2021) aims to allow compositional transfer over a wider range of fine-grained stylistic constructs, including lexical, semantic, stylistic and thematic transfers.

CommonGen Lin et al. (2019) benchmark proposes the task of constrained text generation with generative commonsense reasoning, where given a set of concepts the task is to generate a coherent sentence describing an everyday scenario using the given concepts. To do this successfully, the generative model must reason over commonsense relations between the given concepts (relational reasoning), and infer novel combinations of familiar concepts (compositional generalization). Preliminary analysis shows that current state-of-the-art pre-trained models struggle at the task and generate implausible sentences by a large margin. Other benchmarks proposed in the literature focus on avoiding model hallucinations and assessing the veracity and factuality of current models Hendrycks et al. (2020), Bhakthavatsalam et al. (2021), Talmor et al. (2019). TruthfulQA Lin et al. (2022) benchmark is proposed for measuring the factual accuracy and truthfulness of QA systems. Surprisingly, in their preliminary experiments the authors find that larger language models are less truthful than smaller language models from the same family, neverthleless they are more informative. RealToxicityPrompts Gehman et al. (2020) aims to measure the extent to which toxic degeneration of large language models can be avoided, and the effectiveness of steering text generation algorithms away from producing racist, sexist and toxic content. Ideally, we want to have NLG models that are controllable, truthful, informative and perform well in the real world, however current pre-trained large language models can degenerate into toxic texts even from seemingly innocuous prompts. Instruction-Following Eval (IFEval) Zhou et al. (2023) for large language models aims to measure to what extent LLM models can generate texts that satisfy specific lexical, format and utility constraints, such as for example “write in more than 400 words” and “mention the keyword of AI at least 3 times”. However, performance of current models on existing benchmarks is not necessarily representative of their real-world performance. This issue is amplified by the use of biased automated evaluators, for example towards models that generate longer outputs Li et al. (2023a), Dubois et al. (2024a). The research community not only needs better evaluation metrics (as outlined in Section 6), but also better benchmarks. Given the fragility of current NLG benchmarking practices, fallacious interpretations can be derived Dehghani et al. (2021). To minimize the discrepancy between model performance on a given benchmark and its actual usefulness when deployed in real-life situations, benchmarks used for assessing the capabilities of current NLG systems should accurately reflect the end task of interest, as well as the wide diversity of scenarios and constraints encountered in practice. Motivated by the observation that new advances in metrics and models should more directly inform and benefit each other, bidimensional leaderboards Kasai et al. (2022) are proposed to track progress in both generative models and evaluation metrics for constrained text generation tasks such as machine translation, text summarization and image captioning. Nevertheless, a larger issue in terms of natural language evaluation is the gap between how humans use language in the real world, and what current benchmarks can measure Zellers et al. (2021). In addition, many datasets are not an effective indicator of model generalization and real world performance, particularly in the presence of overlap between the train and test sets, leading to inflated evaluation results Elangovan et al. (2021), Dodge et al. (2021). Besides, since massive web-based datasets used to train large language models are often "contaminated" with downstream test sets, it is important to conduct in-depth analyses to disentangle genuine progress in natural language understanding/generalization from rote memorization Magar & Schwartz (2022); overlooking the impact of pre-training data can result in misleading interpretations of model performance Razeghi et al. (2022). Finally, we would like to draw attention on the lack of resources (datasets and evaluation metrics) for many languages other than English.

8 Discussion and Open Challenges

In what follows we review the main challenges associated with constrained NLG outlining why these challenges have not been solved yet, and present the most promising research directions.

In our view, constrained text generation is a more difficult problem compared to other instances of text generation. The difficulty arises from a multitude of factors, including lack of model expressiveness which makes it difficult for current models to incorporate constraints into the objective function, lack of suitable evaluation metrics to assess the extent to which constraints are satisfied (which becomes even more challenging in the presence of multiple constraints), difficulty in the constrained optimization of non-differentiable reward functions, and finally lack of constrained text generation datasets that are illustrative of a wide diversity of constraints. Due to these pressing issues, constrained text generation remains an open challenge in the research community. Advancing the state-of-the-art requires considerable more collective and focused effort. Below we identify the most promising directions for advancing the state-of-the-art for safe and robust constrained NLG.

Multiple constraint satisfaction Most approaches proposed for constrained text satisfaction focus on generating sentences that meet one single desired constraint, nevertheless generating sequences that simultaneously satisfy multiple lexical constraints is an important open research problem in text generative models Liu et al. (2019a), Latif et al. (2020), Hsieh et al. (2021). While incorporating one constraint is already hard enough due to lack of model expressiveness, incorporating multiple constraints poses significant challenges in terms of defining the loss function accounting for all the desired constraints, difficulty in optimizing it and evaluating whether each constraint is satisfied. Approaches that convert the multiple constraint satisfaction problem into allowing the inclusion of pre-specified lexical constraints at decoding time are not optimal either: on the one hand, decoding complexity increases exponentially or linearly in the number of constraints, and on the other hand forcing constraints at every step of the generation process impacts the quality and naturalness of generated texts Post & Vilar (2018). Moreover, many model architectures are designed for sequential sentence generation only (vs. non-monotonic text generation) and it is non-trivial to impose decoding time constraints while maintaining optimal text generation quality Miao et al. (2019).

Dynamically defined constraints Current approaches to constrained text generation assume there is prior knowledge of the constrained textual attributes and the finite set of values these attributes can take on. Nevertheless, there are situations when it may be desirable to impose constraints dynamically, for eg. in conversational systems depending on the system user's statements, reactions and emotions. When dynamically defining constraints, the main challenges are the lack of model expressiveness and robust ways to evaluate whether these constraints are satisfied. In the literature, controling the realization of a sentence based on another's sentence syntax and semantics is a less explored setting for constrained text generation with dynamic constraints which does not require prior knowledge of all the values the control variable might take on Chen et al. (2019). Disentangled latent space representations of syntax and semantics are essential for the manipulation sentence attributes in tasks such as unsupervised paraphrase generation and syntax-transfer generation Bao et al. (2019).

Generative reasoning Current large-scale text generation models display impressive ability to generate fluent texts, nevertheless composing realistically plausible sentences in the presence of constraints remains a significant open challenge. This is illustrative of all challenges associated with constrained text generation, including lack of model expressiveness, lack of suitable evaluation metrics, difficulty in constrained optimization and lack of constrained text generation datasets. Endowing generative models with commonsense reasoning abilities is an important milestone towards advancing machine understanding and intelligence. In general, the great majority of models proposed in the literature only exploit superficial cues via self-attention to solve NLP tasks, without relying on syntactic information or complex reasoning Pham et al. (2021).

Attribute specific datasets The lack of annotated datasets for attribute specific text generation constitutes a bottleneck in the development and adaptation of models for tasks that require fine-grained control over style and topics. For example, in dialogue systems the absence of attribute annotated conversational datasets that can be used for fine-tuning large scale pre-trained models limits control over the generated responses for a desired attribute Madotto et al. (2020). Moreover, such attribute annotated datasets can help with the personalization of dialogue systems, make dialogues safe, supportive and engaging Serban et al. (2015), Zhang et al. (2018a). Personalized dialogue agents that display consistent personalities and viewpoints overcome the unsatisfying experience of a persona-free chit-chat model. Nevertheless, imposing conversational goals on a dialogue agent for learning target-guided strategies requires keyword-augmented conversation datasets for learning how to steer the conversation towards a designated target subject Tang et al. (2019).

Rule constraints While most research that is currently trying to address constrained text generation is focusing on the incorporation of pre-defined utility or lexical constraints, the satisfaction of rule based constraints is equally relevant, particularly when used to define format and syntactic conditions on the output. However, the lack of model expressiveness makes it challenging to incorporate rule based constraints into the loss function at training time. We encourage more effort in this direction likely to open a plethora of new possibilities in how constraints are specified, incorporated and satisfied in models particularly designed for constrained neural text generation.

Evaluation of constrained text generation In general, evaluation of text generative models is an open challenge. The field is missing robust automated evaluation metrics that correlate with human judgements across multiple dimensions of text quality. Evaluation of models for constrained text generation is currently done using the same flawed existing metrics commonly used in unconditional and conditional text generation evaluation, or in an informal way often times in the absence of a rigorous evaluation procedure. Human evaluation remains the gold standard way to assess text quality, however designing evaluation metrics tailored specifically at assessing whether generated texts meet desired constraints altogether with new benchmark datasets for the evaluation of constrained sequence generation are important next steps Latif et al. (2020).

Adversarial Attacks Adversarial examples exploit vulnerabilities in text generation models and represent an active research area. Adversarial triggers in the form of input-agnostic sequences of tokens concatenated to any input dataset can trigger a pre-trained language models to produce biased, racist and discriminatory outputs even when these models are carefully fine-tuned and optimized against adversarial triggers Wallace et al. (2019). Gradient-based adversarial trigger phrase search techniques are used to generate input prompts to a pre-language model that induce biases in the generated output and allows to study strategies for bias mitigation Sheng et al. (2020). Constrained text generation models that are robust to adversarial attacks are needed for the beneficial use of machine learning and artificial intelligence technology in real world applications, as well as to mitigate any potential societal harms and biases associated with the deployment of large pre-trained language models.

While the above directions outline some of the most pressing research challenges associated with constrained text generation, it is nevertheless a non-exhaustive list of all research problems that need increased attention. Other important open challenges include the use of constrained text generation for personalized agents in a wide variety of contexts, such as in dialogue settings Zhang et al. (2018a), and new benchmark datasets that are reflective of real-world constraints for both training/fine-tuning and evaluating constrained text generation models.

9 Conclusion

In this work, we have presented the reasons why constrained natural language generation is an important, yet highly challenging and largely unsolved research problem. Our first contribution consists in clarifying the difference between the ambiguous use of unconditional, conditional and constrained terms in the natural language generation literature, and draw clear boundaries between these concepts by exemplifying instances of natural language generation tasks with their associated conditions and constraints. Among different paradigms of text generation, we consider constrained text generation to be particularly challenging (if not the most challenging), yet also extremely useful. We identify general reasons why constrained natural language generation deserves significant more attention in the research community, including the lack of model expressiveness in incorporating constraints into the objective function at training time, difficulty in constrained optimization algorithms, the lack of suitable evaluation metrics for robustly assessing, comparing model outputs and claiming success in constrained natural language generation, as well as the lack of constrained text generation datasets that are representative of a wide range of real-world constraints for training and fine-tuning these models. We then survey a representative body of recent literature on constrained text generation using neural networks, presenting the main approaches and methods used, as well as their limitations. Our work serves as an informative guide for both researchers and practitioners to become familiar with the current methodology and main challenges, as well as an advocate for advancing the state-of-the-art in constrained NLG. We invite future work in solving the outlined challenges for better, useful, safer, robust constrained text generation and evaluation.

References

Jacopo Amidei, Paul Piwek, and Alistair Willis. Identifying annotator bias: A new irt-based method for bias identification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4787–4797, 2020.

Afra Amini, Tim Vieira, and Ryan Cotterell. Variational best-of-n alignment. *arXiv preprint arXiv:2407.06057*, 2024.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 936–945, 2017.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021.

Kevin Bache, David Newman, and Padhraic Smyth. Text-based measures of document diversity. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 23–31. ACM, 2013.

Ashutosh Baheti, Alan Ritter, Jiwei Li, and William B Dolan. Generating more interesting responses in neural conversation models with distributional constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3970–3980, 2018.

Michiel Bakker, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, et al. Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in Neural Information Processing Systems*, 35:38176–38189, 2022.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. Constrained decoding for neural nlg from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 831–844, 2019.

Sriram Balasubramanian, Naman Jain, Gaurav Jindal, Abhijeet Awasthi, and Sunita Sarawagi. What's in a name? are bert named entity representations just as good for any other name? *arXiv preprint arXiv:2007.06897*, 2020.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6008–6019, 2019.

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

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. Think you have solved direct-answer question answering? try arc-da, the direct-answer ai2 reasoning challenge. *arXiv preprint arXiv:2102.03315*, 2021.

Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

Antoine Bosselut, Asli Çelikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. Discourse-aware neural rewards for coherent text generation. In *NAACL-HLT*, 2018.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, 2016.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Elia Bruni and Raquel Fernández. Adversarial evaluation for open-domain dialogue generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 284–288, 2017.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluation the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 561–570, 2022.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

Arun Tejasvi Chaganty, Stephen Mussman, and Percy Liang. The price of debiasing automatic metrics in natural language evaluation. *arXiv preprint arXiv:1807.02202*, 2018.

William Chan, Nikita Kitaev, Kelvin Guu, Mitchell Stern, and Jakob Uszkoreit. Kermit: Generative insertion-based modeling for sequences. *arXiv preprint arXiv:1906.01604*, 2019.

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*, 2024.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5972–5984, 2019.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1657–1668, 2017.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. On the weaknesses of reinforcement learning for neural machine translation. *arXiv preprint arXiv:1907.01752*, 2019.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pp. 4299–4307, 2017.

Elizabeth Clark, Yangfeng Ji, and Noah A Smith. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2250–2260, 2018.

Elizabeth Clark, Asli Celikyilmaz, and Noah A Smith. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2748–2760, 2019.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith. All that's 'human'is not gold: Evaluating human evaluation of generated text. 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)*, pp. 7282–7296, 2021.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12: 283–298, 2024.

Pierre Jean A Colombo, Chloé Clavel, and Pablo Piantanida. Infolm: A new metric to evaluate summarization & data2text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10554–10562, 2022.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, et al. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*, 2016.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

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

Mostafa Dehghani, Yi Tay, Alexey A Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. The benchmark lottery. *arXiv preprint arXiv:2107.07002*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1286–1305, 2021.

Chris Donahue, Mina Lee, and Percy Liang. Enabling language models to fill in the blanks. *arXiv preprint arXiv:2005.05339*, 2020.

Yi Dong, Zhilin Wang, Makesh Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 11275–11288, 2023.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024a.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024b.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2836–2846, 2020.

Aparna Elangovan, Jiayuan He, and Karin Verspoor. Memorization vs. generalization: Quantifying data leakage in nlp performance evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1325–1335, 2021.

Allyson Ettinger. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48, 2020.

Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2214–2220, 2019.

Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 45–54, 2018a.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018b.

Angela Fan, Mike Lewis, and Yann Dauphin. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2650–2660, 2019.

William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: Better text generation via filling in the __. *arXiv preprint arXiv:1801.07736*, 2018.

Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. *EMNLP 2017*, pp. 94, 2017.

Rudolf Franz Flesch. *How to write plain English: A book for lawyers and consumers*. Harpercollins, 1979.

Marina Fomicheva and Lucia Specia. Taking mt evaluation metrics to extremes: Beyond correlation with human judgments. *Computational Linguistics*, 45(3):515–558, 2019.

Markus Freitag, David Grangier, and Isaac Caswell. Bleu might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 61–71, 2020.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*, 2023.

Yao Fu and Yansong Feng. Natural answer generation with heterogeneous memory. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 185–195, 2018.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. deltableu: A discriminative metric for generation tasks with intrinsically diverse targets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pp. 445–450, 2015.

Jianfeng Gao, Michel Galley, and Lihong Li. *Neural approaches to conversational AI: Question answering, task-oriented dialogues and social chatbots*. Now Foundations and Trends, 2019a.

Yang Gao, Christian M Meyer, Mohsen Mesgar, and Iryna Gurevych. Reward learning for efficient reinforcement learning in extractive document summarisation. *arXiv preprint arXiv:1907.12894*, 2019b.

Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

Pablo Gervás. Computational approaches to storytelling and creativity. *AI Magazine*, 30(3):49–49, 2009.

Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1183–1191, 2016.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, pp. 43–48, 2017.

Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. Affect-lm: A neural language model for customizable affective text generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 634–642, 2017.

Kevin Gimpel, Dhruv Batra, Chris Dyer, and Gregory Shakhnarovich. A systematic exploration of diversity in machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1100–1111, 2013.

Dimitra Gkatzia and Saad Mahamood. A snapshot of nlg evaluation practices 2005-2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pp. 57–60, 2015.

Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. Plan, write, and revise: an interactive system for open-domain story generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 89–97, 2019.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Yvette Graham, Barry Haddow, and Philipp Koehn. Translationese in machine translation evaluation. *arXiv preprint arXiv:1906.09833*, 2019.

Jiatao Gu, Qi Liu, and Kyunghyun Cho. Insertion-based decoding with automatically inferred generation order. *arXiv preprint arXiv:1902.01370*, 2019a.

Jiatao Gu, Changhan Wang, and Jake Zhao. Levenshtein transformer. *arXiv preprint arXiv:1905.11006*, 2019b.

Lin Gui, Cristina Gârbacea, and Victor Veitch. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *arXiv preprint arXiv:2406.00832*, 2024.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. *arXiv preprint arXiv:2401.07453*, 2024.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association of Computational Linguistics*, 6:437–450, 2018.

Helena H. Lee, Ke Shu, Palakorn Achananuparp, Philips Kokoh Prasetyo, Yue Liu, Ee-Peng Lim, and Lav R Varshney. Recipegpt: Generative pre-training based cooking recipe generation and evaluation system. In *Companion Proceedings of the Web Conference 2020*, pp. 181–184, 2020.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. Warp: Word-level adversarial reprogramming. 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)*, pp. 4921–4933, 2021.

Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji. Lm-switch: Lightweight language model conditioning in word embedding space. *arXiv preprint arXiv:2305.12798*, 2023.

Tatsunori B Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation. *arXiv preprint arXiv:1904.02792*, 2019.

Tianxing He, Jingyu Zhang, Tianle Wang, Sachin Kumar, Kyunghyun Cho, James Glass, and Yulia Tsvetkov. On the blind spots of model-based evaluation metrics for text generation. *arXiv preprint arXiv:2212.10020*, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. Measuring and manipulating knowledge representations in language models. *arXiv preprint arXiv:2304.00740*, 2023.

Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1535–1546, 2017.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1638–1649, 2018.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Lee-Hsun Hsieh, Yang-Yin Lee, and Ee-Peng Lim. Enconter: Entity constrained progressive sequence generation via insertion-based transformer. *arXiv preprint arXiv:2103.09548*, 2021.

J Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. Improved lexically constrained decoding for translation and monolingual rewriting. 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)*, pp. 839–850, 2019.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1587–1596. JMLR. org, 2017.

EunJeong Hwang, Bodhisattwa Majumder, and Niket Tandon. Aligning language models to user opinions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5906–5919, 2023.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Daphne Ippolito, Reno Kriz, Joao Sedoc, Maria Kustikova, Chris Callison-Burch, Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, Chris Callison-Burch, John Hewitt, et al. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. Association for Computational Linguistics, 2018.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 633–644, 2014.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1875–1885, 2018.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity?a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *EMNLP 2017*, 6:10, 2017.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 424–434, 2019.

Samia Kabir, David N Udo-Imeh, Bonan Kou, and Tianyi Zhang. Who answers it better? an in-depth analysis of chatgpt and stack overflow answers to software engineering questions. *arXiv preprint arXiv:2308.02312*, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Tomoyuki Kajiwara. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6047–6052, 2019.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pp. 15696–15707. PMLR, 2023.

Anjuli Kannan and Oriol Vinyals. Adversarial evaluation of dialogue models. *arXiv preprint arXiv:1701.08198*, 2017.

Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah A Smith. Transparent human evaluation for image captioning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3464–3478, 2022.

Pei Ke, Hao Zhou, Yankai Lin, Peng Li, Jie Zhou, Xiaoyan Zhu, and Minlie Huang. Ctrleval: An unsupervised reference-free metric for evaluating controlled text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2306–2319, 2022.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

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

Mitesh M Khapra and Ananya B Sai. A tutorial on evaluation metrics used in natural language generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pp. 15–19, 2021.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In *EMNLP*, 2016.

Yoon Kim, Kelly Zhang, Alexander M Rush, Yann LeCun, et al. Adversarially regularized autoencoders for generating discrete structures. *arXiv preprint arXiv:1706.04223*, 2:12, 2017.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975.

Rebecca Knowles and Philipp Koehn. Neural interactive translation prediction. In *Twelfth Conference of The Association for Machine Translation in the Americas*, pp. 107–120. Association for Machine Translation in the Americas, AMTA, 2016.

Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. Aligning large language models with representation editing: A control perspective. *arXiv preprint arXiv:2406.05954*, 2024.

Xiang Kong, Bohan Li, Graham Neubig, Eduard Hovy, and Yiming Yang. An adversarial approach to high-quality, sentiment-controlled neural dialogue generation. *arXiv preprint arXiv:1901.07129*, 2019.

Bernhard Kratzwald, Anna Eigenmann, and Stefan Feuerriegel. Rankqa: Neural question answering with answer re-ranking. *arXiv preprint arXiv:1906.03008*, 2019.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. Gedi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367*, 2020.

Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1777–1788, 2018.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. 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)*, pp. 540–551, 2019.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pp. 166–172, 2019.

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

Philippe Laban, Andrew Hsi, John Canny, and Marti A Hearst. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, volume 1, 2020.

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

Seemab Latif, Sarmad Bashir, Mir Muntasar Ali Agha, and Rabia Latif. Backward-forward sequence generative network for multiple lexical constraints. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pp. 39–50. Springer, 2020.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1173–1182, 2018.

Christoph Leiter, Piyawat Lertvittayakumjorn, Marina Fomicheva, Wei Zhao, Yang Gao, and Steffen Eger. Towards explainable evaluation metrics for natural language generation. *arXiv preprint arXiv:2203.11131*, 2022.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL-HLT*, pp. 110–119, 2016a.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and William B Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 994–1003, 2016b.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, 2016c.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2157–2169, 2017.

Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1865–1874, 2018.

Junyi Li, Charith Peris, Ninareh Mehrabi, Palash Goyal, Kai-Wei Chang, Aram Galstyan, Richard Zemel, and Rahul Gupta. The steerability of large language models toward data-driven personas. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7283–7298, 2024a.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*, 2022a.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024b.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022b.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023a.

Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. Guiding large language models via directional stimulus prompting. *arXiv preprint arXiv:2302.11520*, 2023b.

Zichao Li, Ines Arous, Siva Reddy, and Jackie Chi Kit Cheung. Evaluating dependencies in fact editing for language models: Specificity and implication awareness. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7623–7636, 2023c.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Bill Yuchen Lin, Ming Shen, Yu Xing, Pei Zhou, and Xiang Ren. Commongen: A constrained text generation dataset towards generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*, 2019.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*, 2021.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2122–2132, 2016.

Dayiheng Liu, Jie Fu, Qian Qu, and Jiancheng Lv. Bfgan: Backward and forward generative adversarial networks for lexically constrained sentence generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):2350–2361, 2019a.

Jiateng Liu, Pengfei Yu, Yuji Zhang, Sha Li, Zixuan Zhang, and Heng Ji. Evedit: Event-based knowledge editing with deductive editing boundaries. *arXiv preprint arXiv:2402.11324*, 2024.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023a.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 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)*, pp. 9031–9041, 2020.

Xin Liu, Muhammad Khalifa, and Lu Wang. Bolt: Fast energy-based controlled text generation with tunable biases. *arXiv preprint arXiv:2305.12018*, 2023b.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023c.

Yu Lu Liu, Rachel Bawden, Thomas Scaliom, Benoît Sagot, and Jackie Chi Kit Cheung. Maskeval: Weighted mlm-based evaluation for text summarization and simplification. *arXiv preprint arXiv:2205.12394*, 2022.

Zhiqiang Liu, Zuohui Fu, Jie Cao, Gerard de Melo, Yik-Cheung Tam, Cheng Niu, and Jie Zhou. Rhetorically controlled encoder-decoder for modern chinese poetry generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1992–2001, 2019b.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pp. 5103–5113, 2018.

Ekaterina Loginova, Stalin Varanasi, and Günter Neumann. Towards multilingual neural question answering. In *European Conference on Advances in Databases and Information Systems*, pp. 274–285. Springer, 2018.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4288–4299, 2021.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pp. 700–709, 2018.

Yiwei Lyu, Paul Pu Liang, Hai Pham, Eduard Hovy, Barnabás Póczos, Ruslan Salakhutdinov, and Louis-Philippe Morency. Styleptb: A compositional benchmark for fine-grained controllable text style transfer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2116–2138, 2021.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabás Póczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1869–1881, 2020.

Andrea Madotto, Etsuko Ishii, Zhaojiang Lin, Sumanth Dathathri, and Pascale Fung. Plug-and-play conversational models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 2422–2433, 2020.

Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 157–165, 2022.

Louis Martin, Éric Villemonte de la Clergerie, Benoît Sagot, and Antoine Bordes. Controllable sentence simplification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pp. 4689–4698, 2020.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 14867–14875, 2021.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2799–2808, 2019.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, 2020.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Coherent dialogue with attention-based language models. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022a.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022b.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6834–6842, 2019.

Yishu Miao and Phil Blunsom. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 319–328, 2016.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. Mix and match: Learning-free controllable text generationusing energy language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 401–415, 2022.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

Lili Mou, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. Backward and forward language modeling for constrained sentence generation. *arXiv preprint arXiv:1512.06612*, 2015.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970*, 2016.

Jonas Mueller, David Gifford, and Tommi Jaakkola. Sequence to better sequence: continuous revision of combinatorial structures. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2536–2544. JMLR. org, 2017.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 588–593, 2015.

Nikola I Nikolov and Richard Hahnloser. Abstractive document summarization without parallel data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pp. 6638–6644, 2020.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 260–266, 2019.

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

Hugo Gonçalo Oliveira. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, pp. 11–20, 2017.

OpenAI. Introducing chatgpt. *https://openai.com/blog/chatgpt*, 2022.

OpenAI. Gpt-4 technical report, 2023.

Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. Editing implicit assumptions in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7053–7061, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4812–4829, 2021.

Alexander Pan, Jun Shern Chan, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark. In *International Conference on Machine Learning*, pp. 26837–26867. PMLR, 2023.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*, 2024.

Ramakanth Pasunuru, Han Guo, and Mohit Bansal. Dorb: Dynamically optimizing multiple rewards with bandits. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7766–7780, 2020.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

Jonathan Pei, Kevin Yang, and Dan Klein. Preadd: Prefix-adaptive decoding for controlled text generation. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

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

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 74–84, 2017.

Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1145–1160, 2021.

Maja Popović. On reducing translation shifts in translations intended for mt evaluation. In *Proceedings of Machine Translation Summit XVII: Translator, Project and User Tracks*, pp. 80–87, 2019.

Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1314–1324, 2018.

Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5046–5056, 2019.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551, 2022.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36, 2024.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Sudha Rao and Joel Tetreault. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 129–140, 2018.

Ankit Singh Rawat, Chen Zhu, Daliang Li, Felix Yu, Manzil Zaheer, Sanjiv Kumar, and Srinadh Bhojanapalli. Modifying memories in transformer models. In *International Conference on Machine Learning (ICML)*, volume 2020, 2021.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*, 2022.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2685–2702, 2020.

Machel Reid and Graham Neubig. Learning to model editing processes. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3822–3832, 2022.

Ehud Reiter and Anja Belz. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558, 2009.

Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–7, 2021.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

Domenic Rosati, Robie Gonzales, Jinkun Chen, Xuemin Yu, Melis Erkan, Yahya Kayani, Satya Deepika Chavatapalli, Frank Rudzicz, and Hassan Sajjad. Long-form evaluation of model editing. *arXiv preprint arXiv:2402.09394*, 2024.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E Peters, and Matt Gardner. Tailor: Generating and perturbing text with semantic controls. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pp. 3194–3213. Association for Computational Linguistics (ACL), 2022.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, pp. 5228–5237, 2018.

Abdelrhman Saleh, Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, and Rosalind Picard. Hierarchical reinforcement learning for open-domain dialog. *arXiv preprint arXiv:1909.07547*, 2019.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In *International Conference on Machine Learning*, pp. 29971–30004. PMLR, 2023.

Stephanie Schoch, Diyi Yang, and Yangfeng Ji. "this is a problem, don't you agree?" framing and bias in human evaluation for natural language generation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*, pp. 10–16, 2020.

Moritz Schubotz, Philipp Scharpf, Kaushal Dudhat, Yash Nagar, Felix Hamborg, and Bela Gipp. Introducing mathqa: a math-aware question answering system. *Information Discovery and Delivery*, 2018.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, 2017.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of NAACL-HLT*, pp. 1702–1723, 2019.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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*, pp. 35–40, 2016.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*, 2015.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Lei Sha. Gradient-guided unsupervised lexically constrained text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8692–8703, 2020.

Darsh J Shah, Tal Schuster, and Regina Barzilay. Automatic fact-guided sentence modification. In *AAAI*, pp. 8791–8798, 2020.

Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*, 2023.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pp. 6830–6841, 2017.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3407–3412, 2019.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. Towards controllable biases in language generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 3239–3254, 2020.

Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. Toward diverse text generation with inverse reinforcement learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4361–4367. AAAI Press, 2018.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. Ruse: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 751–758, 2018.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. On the effect of pretraining corpora on in-context learning by a large-scale language model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5168–5186, 2022.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235, 2020.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*, 2022.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*, 2023.

Koustuv Sinha, Prasanna Parthasarathi, Jasmine Wang, Ryan Lowe, William L Hamilton, and Joelle Pineau. Learning an unreferenced metric for online dialogue evaluation. *arXiv preprint arXiv:2005.00583*, 2020.

Amanda Stent, Matthew Marge, and Mohit Singhai. Evaluating evaluation methods for generation in the presence of variation. In *international conference on intelligent text processing and computational linguistics*, pp. 341–351. Springer, 2005.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. Insertion transformer: Flexible sequence generation via insertion operations. In *International Conference on Machine Learning*, pp. 5976–5985, 2019.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124*, 2022.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. Transforming delete, retrieve, generate approach for controlled text style transfer. *arXiv preprint arXiv:1908.09368*, 2019.

Elior Sulem, Omri Abend, and Ari Rappoport. Bleu is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 738–744, 2018.

Simeng Sun, Dhawal Gupta, and Mohit Iyyer. Exploring the impact of low-rank adaptation on the performance, efficiency, and regularization of rlhf. *arXiv preprint arXiv:2309.09055*, 2023.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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)*, pp. 4149–4158, 2019.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. Evaluating the factual consistency of large language models through summarization. *arXiv preprint arXiv:2211.08412*, 2022.

Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu. Target-guided open-domain conversation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5624–5634, 2019.

Zecheng Tang, Pinzheng Wang, Keyan Zhou, Juntao Li, Ziqiang Cao, and Min Zhang. Can diffusion model achieve better performance in text generation? bridging the gap between training and inference! *arXiv preprint arXiv:2305.04465*, 2023.

L Theis, A van den Oord, and M Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR 2016)*, pp. 1–10, 2016.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Alex Turner, M Monte, David Udell, Lisa Thiergart, and Ulisse Mini. Steering gpt-2-xl by adding an activation vector. In *AI Alignment Forum*, 2023.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 355–368, 2019.

Judith van Stegeren and Mariët Theune. Narrative generation in the wild: Methods from nanogenmo. In *Proceedings of the Second Workshop on Storytelling*, pp. 65–74, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. 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)*, 2019.

Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. Steering output style and topic in neural response generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2140–2150, 2017.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*, 2024a.

Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*, 2023a.

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

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36:74764–74786, 2023b.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*, 2024b.

Sean Welleck, Kianté Brantley, Hal Daumé III, and Kyunghyun Cho. Non-monotonic sequential text generation. *arXiv preprint arXiv:1902.02192*, 2019.

Georg Wiese, Dirk Weissenborn, and Mariana Neves. Neural domain adaptation for biomedical question answering. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 281–289, 2017.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. Beyond bleu: Training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4344–4355, 2019.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*, 2017.

Yuxiang Wu and Baotian Hu. Learning to extract coherent summary via deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*, 2023.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 1015–1024. Association for Computational Linguistics, 2012.

Joern Wuebker, Spence Green, John DeNero, Saša Hasan, and Minh-Thang Luong. Models and inference for prefix-constrained machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, 2016.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. Topic augmented neural response generation with a joint attention mechanism. *arXiv preprint arXiv:1606.08340*, 2(2), 2016.

Kevin Yang and Dan Klein. Fudge: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535, 2021.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

Nai-Chi Yang, Wei-Yun Ma, and Pu-Jen Cheng. Plug-in language model: Controlling text generation with a simple regression model. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2165–2181, 2024a.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*, 2024b.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34, 2021.

Rowan Zellers, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi. Turingadvice: A generative and dynamic evaluation of language use. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 521–530, 2016.

Hanqing Zhang and Dawei Song. Discup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3392–3406, 2022.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 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)*, pp. 2204–2213, 2018a.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

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

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*, pp. 1810–1820, 2018b.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. Pointer: Constrained text generation via insertion-based generative pre-training. *arXiv preprint arXiv:2005.00558*, 2020.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*, 2018c.

Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M Rush, and Yann LeCun. Adversarially regularized autoencoders. In *35th International Conference on Machine Learning, ICML 2018*, pp. 9405–9420. International Machine Learning Society (IMLS), 2018.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. 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)*, pp. 563–578, 2019.

Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. Personalized dialogue generation with diversified traits. *arXiv preprint arXiv:1901.09672*, 2019.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*, 2023.

Chunting Zhou and Graham Neubig. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 310–320, 2017.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1393–1404, 2021.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1097–1100. ACM, 2018.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. Controllable generation from pre-trained language models via inverse prompting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2450–2460, 2021.