PCFG-Based Natural Language Interface Improves Generalization for Controlled Text Generation

Jingyu Zhang
Johns Hopkins University
jzhan237@jhu.edu

James Glass
MIT
glass@mit.edu

Tianxing He
University of Washington
goosehe@cs.washington.edu

Abstract

Existing work on controlled text generation (CTG) assumes a control interface of categorical attributes. In this work, we propose a natural language (NL) interface, where we craft a PCFG to embed the control attributes into natural language commands, and propose variants of existing CTG models that take commands as input. In our experiments, we design tailored setups to test the model's generalization abilities. We find our PCFG-based command generation approach is effective for handling unseen commands compared to fix-set templates. Further, our proposed NL models can effectively generalize to unseen attributes (a new ability enabled by the NL interface), as well as unseen attribute combinations. Interestingly, in model comparisons, the simple conditional generation approach, enhanced with our proposed NL interface, is shown to be a strong baseline in those challenging settings.

1 Introduction

With the advancement of large-scale pretraining, language models (LM) are now able to generate increasingly more realistic text (Radford et al., 2019; Brown et al., 2020; Rae et al., 2021; Hoffmann et al., 2022; Smith et al., 2022; Thoppilan et al., 2022). Therefore, how to control the generation of LMs has become an important research topic. In *controlled text generation* (CTG), a series of works (Keskar et al., 2019; Dathathri et al., 2020; Krause et al., 2021; Yang and Klein, 2021; Liu et al., 2021; Yu et al., 2021; Li et al., 2022, *inter alia*) propose model frameworks to generate text conditioned on some desired (user-specified) attribute a. These attributes, which depend on the datasets of interest, could be topic, formality, sentiment, etc.

An important assumption behind this controlled generation setting is that the attributes are chosen from **a fixed set** (i.e., they are treated as categorical random variables). Although this setting is convenient, it seriously limits the applications of the

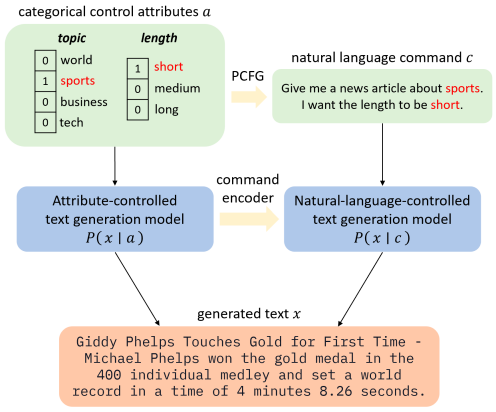

Figure 1: We explore generation models that take natural language commands as input. For training, we use PCFG to embed categorical control attributes into natural language commands.

CTG system: (1) Since the attribute set is fixed during training, it would be impossible for the model to generalize to unseen options if used as-is. (2) This interface is not very human-friendly, because it could be difficult for users to navigate through the (possibly long) lists of options. Motivated by these limitations, in this work we propose a *natural language interface* for CTG, illustrated in Figure 1. With this change of interface, the input to the CTG model changes from one-hot vectors to natural language commands (for short, *commands*). To efficiently train this system and enable it to generalize, we design a probabilistic context-free grammar (PCFG) to embed categorical attributes into a diverse set of natural language commands.

Using natural language instruction has been explored in recent work (Sanh et al., 2021; Wei et al., 2022; Mishra et al., 2022; Reif et al., 2022; Schick and Schütze, 2021). Our work differs from theirs in (1) We focus on the task of CTG as opposed to the performance on cross-task generalization, and design tailored scenarios for evaluation. (2) We introduce PCFG for command generation, which has

*Proceedings of the The 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 295–313
July 13-14, 2023

not been explored by previous work. We discuss this relationship in more detail in Section 2.

The change of interface brings several immediate benefits: (1) Natural language inputs enable the system to generalize to unseen attribute options (as long as they can be expressed in natural language). (2) Unlike fixed-set template sentences in previous works, the PCFG can generate diverse natural language variation during training, which we will show is crucial for generalization. (3) The input process becomes more natural and interactive to a human user, and it can be linked with, for example, a speech recognition module.

With this new interface, we propose variants of several existing CTG systems that take commands as input, and design experiments to compare different CTG models under tailored scenarios. We briefly summarize our main contributions below:

- We propose a PCFG-based natural language interface for controlled text generation. The natural language interface enables zero-shot generalization on control attributes unseen during training, a capability previously impossible due to the fixed-set assumption.
- We show that training with commands generated by a PCFG is an effective method for increasing natural language variation over using fixed-set templates, allowing natural language CTG models to better generalize to commands unseen during training.
- We test the proposed natural language CTG models on settings where the models need to generalize to unseen attributes and attribute combinations. Surprisingly, the simple conditional generation approach is shown to be a strong baseline in these challenging setups.

2 Related Work

Controlled Text Generation In open-ended text generation, a series of approaches have been proposed to control the generation to satisfy certain attributes (e.g. topic) (Keskar et al., 2019; Dathathri et al., 2020; Krause et al., 2021; Yang and Klein, 2021; Liu et al., 2021, *inter alia*). Some of these studies utilize a trained classifier to guide the generative model towards the desired attribute, while others use a smaller LM to reweight LM logits. Very recently, Li et al. (2022) focus on controlling more complex attributes such as syntactic structure with a non-autoregressive LM. Another line of work conducts CTG via prompt learning (Clive et al., 2022; Yang et al., 2022). These work assume a fixed set of control attributes.

Our NL interface is more related to Yu et al. (2021), which uses an attribute alignment function to embed attribute words into a hidden representation that guides LM generation. The attribute alignment function does not assume attribute tokens are from a fixed set, so it is possible to do inference on an attribute token not seen in training. Keyword2Text (Pascual et al., 2021) shift the distribution over vocabulary toward words that are semantically similar to control keywords in a discriminator-free manner, thus does not assume a fixed set of keywords. Besides attribute control, lexically constrained decoding (Post and Vilar, 2018) has also been used to enforce certain key phrases to be included in the generation (Mao et al., 2020). Different from these work which uses keywords, we utilize PCFG to construct fully-natural-language sentences as commands.

Instruction Following A recent series of work proposes to describe NLP tasks in natural language, and use the task description as an instruction to promote zero-shot generalization for LMs (Sanh et al., 2021; Wei et al., 2022, *inter alia*). Such task descriptions are manually created, detailed definitions of NLP tasks, which contain explanations about input, output, emphasis, and possibly a small number of demonstrative examples. InstructGPT (Ouyang et al., 2022) uses an RL policy to improve LM's capability to follow user instructions.

Although our work resembles these works in the form of natural language instructions, we note several important differences. First, existing works focus on general instruction following that is applicable to a very broad range of tasks and evaluate on generalization capabilities across tasks. We specifically consider the use of NL commands in the CTG setting and compare variants of CTG models in tailored test scenarios. Moreover, previous works in natural language instruction employ a fixed number of templates for each task, whereas we craft a PCFG that can generate a diverse set of command sentences to serve as templates. We show the effectiveness of our PCFG over fixed-set templates in subsequent experiments in Section 5.1. Finally, prompting models with NL instructions fails for moderately sized LMs without any modifications Li and Liang (2021). Thus, it is non-trivial to adapt NL instruction to smaller models.

3 Framework

The goal of controlled text generation is to model the conditional distribution $P(x|a)$ so that the generated text x satisfies the desired attributes a. a could include multiple attributes (e.g., topic and length), and we will use a_i to denote the ith attribute. In the standard categorical setting, the attribute a_i are from a fixed set of pre-defined options. We assume there are m attributes of interest ($m \leqslant 2$ in our experiments). In the next few sections, we describe the PCFG that we craft to embed the categorical attributes, and our proposed NL variants of several existing CTG systems.

3.1 Embedding Attributes into Commands

We embed categorical attributes into natural language commands with a PCFG.[1] We favor PCFG due to its ability to generate diverse NL variations expressing the same control semantics. For simplicity, most of the probability weights are set to uniform. In this section, we will describe it at the high-level, and more details and the full set of rules are provided in Appendix C. Table 1 is a concrete example of how a command describing an AG news article with a sports topic could be generated by our PCFG. We clarify that while the PCFG is used for training and testing in our work, the end user will not need to use it, as the model can generalize to unseen commands (Section 5.1).

Our command generation has three steps. First, a template with m attribute slots is generated by the PCFG. We design the PCFG to generate templates that "ask" the system to generate text with some attributes and domains. We first sample a top-level seed template from `ROOT` that determines high-level sentence structure (e.g., `[PLS] [HEAD-FORM]` a `[TEXT-FORM] [LABEL-SEG]`), then fill in sentence segments with PCFG rules (e.g., `[HEAD-FORM]` will be substituted by "generate"). These sentence segments are neither domain nor attribute specific and thus can be used regardless of the attributes. In contrast to writing a set of fixed templates, our PCFG has multiple levels of rule and can greatly improve NL variation.

Next, we verbalize the domain media D, attribute a, and attribute name A into natural language by crafting PCFG rules that transform them into words or phrases. Considering the fact that different words could have similar meanings in natural language, these mappings could be one-to-many to further improve NL variation. For instance, news about "business" can also be described as "commerce", and "very negative" is similar to "terrible".

Finally, we conduct a postprocessing step to correct simple grammar errors, e.g., "a AG news article" would be corrected as "an AG news article".

In our preliminary attempts, we attempted to train a conditional neural LM for command generation, instead of using a PCFG. Although the neural model has better diversity, the stochastic nature of sampling makes the attribute embedding inaccurate. Besides, training such a neural LM would require a large amount of (attribute, command) paired data. Therefore we turn to a PCFG approach as it has guaranteed accuracy, with decent diversity.

1. PCFG-based template generation
(1) Generate top-level seed template from `ROOT`:
⇒ `[PLS] [HEAD-FORM]` a `[TEXT-FORM] [LABEL-SEG].`
(2) Select PCFG rules to generate template:
`[PLS]` → . . . → please, `[HEAD-FORM]` → . . . → generate,
`[TEXT-FORM]` → . . . → D
`[LABEL-SEG]` → . . . → with a a A
⇒ please generate a D with a a A.
2. Verbalize
⇒ please generate a AG news report with a sports topic.
3. Postprocess
⇒ Please generate an AG news report with a sports topic.

Table 1: Examples of PCFG command generation. `ROOT` is the PCFG start symbol. Newly replaced segments are highlighted in red. In step 1.(2), we omit intermediate PCFG expansions to "→ . . . →".

3.2 Models

In this section, we first review some existing CTG models. For the new NL interface, we propose natural variants of the models which take commands as input. All models are based on a pretrained autoregressive LM, denoted by P_b.

3.2.1 PrefixLM

A direct method to model the conditional distribution $P(x|a)$ is to encode the attribute as a prefix and finetune the base model to generate x conditioned on the prefix. In the standard categorical attribute setting, we randomly initialize an embedding vector for each attribute and feed the corresponding embeddings as the prefix. Multiple attributes are arranged in a pre-defined order.

PrefixLM-NL The NL variant of PrefixLM is straightforward. We just use the command as the prefix. No extra parameters need to be added.

[1] Note that our command generation process is not strictly a PCFG, but it is very close.

3.2.2 Future Discriminator Controlled Generation (FUDGE)

FUDGE (Yang and Klein, 2021) decomposes the conditional distribution using Bayes' rule according to Equation 1:

$$P_{\text{fudge}}(x_i|x_{1:i-1}, a) \propto P_{\text{b}}(x_i|x_{1:i-1})P_{\text{cls}}(a|x_{1:i}). \tag{1}$$

It involves training a future discriminator to predict whether the generated prefix $x_{1:i}$ will lead to a full generation that satisfies the attribute a. Following FUDGE's original formulation, we assume different attributes are conditionally independent and train a discriminator $P(a_k|x_{1:i})$ for each attribute a_k. We then use their product as the probability that all attributes are satisfied, i.e., $P(a_1, \ldots, a_m|x_{1:i}) = \prod_k P(a_k|x_{1:i})$.

As we consider attributes with multiple options (e.g., 4 topics or 5 sentiments), the FUDGE discriminator for a single attribute is a multiclass classification model that predicts the conditional distribution $P(a|x_{1:i})$ over all possible options of attribute a.

FUDGE-NL In order to enable FUDGE to handle natural-language commands, we utilize a binary alignment discriminator to judge whether the generated text aligns with the command. Given a command c, let $y_c \in \{0, 1\}$ be a binary variable that denotes whether the prefix $x_{1:i}$ aligns with the command. Control is achieved by generating from the conditional distribution $P(x_i|x_{1:i-1}, y_c = 1)$ that the alignment property is satisfied. We modify FUDGE's decomposition as Equation 2:

$$\begin{aligned} &P_{\text{fudge-nl}}(x_i|x_{1:i-1}, y_c = 1) \\ &\quad \propto P_{\text{b}}(x_i|x_{1:i-1})P_{\text{cls}}(y_c = 1|x_{1:i}). \end{aligned} \tag{2}$$

$P_{\text{cls}}(y_c = 1|x_{1:i})$ is modeled by a binary classifier trained on a dataset of command and generation prefix pairs $\{(c, x_{1:i})\}$. To create this data, for a given example text x with attributes a, we first apply our PCFG to generate a true command c^{pos}. We then randomly flip one (or both) of the attribute in a, and generate a false command c^{neg}. By pairing c^{pos} and c^{neg} with x, we obtain the positive/negative training data for the discriminator. In practice, we concatenate the command and generation prefix (separated by a special `[SEP]` token) and feed it as input to the alignment discriminator.

FUDGE-Binary One major difference between FUDGE and its NL variant is that the discriminator is always binary for FUDGE-NL due to the alignment objective. This inspires us to propose a binary variant of the FUDGE model, FUDGE-Binary, which operates with the categorical interface. Similar to FUDGE-NL, we use a binary variable y_a to denote whether $x_{1:i}$ aligns with attribute a, and modify the decomposition as:

$$\begin{aligned} &P_{\text{fudge-bin}}(x_i|x_{1:i-1}, y_a = 1) \\ &\quad \propto P_{\text{b}}(x_i|x_{1:i-1})P_{\text{cls}}(y_a = 1|x_{1:i}). \end{aligned} \tag{3}$$

FUDGE-Binary's discriminator will always make a binary prediction even if there are more than two options for a single attribute. Since attributes are still from a fixed set, we use a single classification model but attach a separate classifier head for each option. During training, the classification head W_{a^*} that matches the correct attribute a^* receives a correct label $y = 1$, and all other classification heads $\{W_a\}_{a \neq a^*}$ receive label $y = 0$. At test time, we select the classification head W_a base on the desired attribute a to predict the alignment probability $P(y_a = 1|x_{1:i})$. Although this variant is a simple modification from the original FUDGE, empirically we find it to achieve stronger performance in the categorical interface.

4 Experimental Setup

4.1 Datasets

We utilize two popular text classification datasets for our experiments: AG News and Yelp Review.[2] For each dataset, we consider two control attributes: label and length. The label attribute is extracted from the classification label, i.e., topic labels for AG News and sentiment labels for Yelp Review. There are 4 topics {world, sports, business, science/tech} in AG News and 5 sentiment classes ranging from most positive to most negative in Yelp Review. The length attribute is created by dividing the dataset to n_{len} length ranges so that number of training examples in each length range is balanced. We use $n_{\text{len}} = 3$ for AG News and $n_{\text{len}} = 5$ for Yelp Review. We refer readers to Appendix A for details about dataset preprocessing.

4.2 Evaluation Metrics

We measure the generation performance in three aspects: control accuracy, quality, and diversity. In our experiments, we find that different variants of models mostly perform comparably on quality or

[2]Obtained from Hugging Face Datasets.

diversity aspects. Therefore, we will mainly focus our discussion on control accuracy.

Control Accuracy To evaluate the effectiveness of the control, we consider three types of control accuracy: LABEL ACCURACY refers to the accuracy that the generation satisfies the classification label, i.e., topic classification accuracy on AG News and sentiment classification accuracy on Yelp. This metric is computed by a RoBERTa classifier finetuned on the corresponding classification dataset. LENGTH ACCURACY refers to the accuracy that the generation's tokenized length lies within the predefined length range. COMPOSITIONAL ACCURACY is the accuracy that both label and length attributes are satisfied.

Text Quality We consider two metrics to measure the quality of the generated text. GPT-NEO PERPLEXITY (G-PPL): we finetune the GPT-Neo-1.3B model[3] on the corresponding datasets (without the labels), and report the perplexity of the generated text given by it. BLEU score: we randomly sample 100 examples from the AG News or Yelp test set as the reference, and compute the 4-gram BLEU score.

Diversity We measure diversity of the generated text using 4-gram TEXT ENTROPY (Zhang et al., 2018). That is, treat the generated token frequency as a discrete distribution, and compute its entropy.

4.3 Model Instantiation

Here we describe the implementation of models mentioned in Section 3.2. We use the Hugging Face transformers library (Wolf et al., 2020) and adapt from FUDGE's released code.[4]

For all models, we produce generation by top-k sampling with $k = 20$ unless otherwise stated.

PrefixLM variants We finetune a GPT-2 (Radford et al., 2019) small model without any modification (except for adding necessary special tokens) for both PrefixLM and PrefixLM-NL. At test time, we feed the desired attributes or command sentences as the prefix and evaluate on the continuation produced by the model.

FUDGE variants The backbone language model P_b for FUDGE models is a GPT-2 small model finetuned on the corresponding dataset, using the same data available at discriminator training. That is, under the zero-shot setting, we use the same data configuration to finetune the backbone LM.

For FUDGE and FUDGE-Binary, we train two discriminator for each of the label (topic or sentiment) and length attribute; FUDGE-NL use a single alignment discriminator to handle commands.

Each discriminator for FUDGE and FUDGE-NL is a GPT-2 small model followed by a single linear classification layer (with different numbers of output classes). The discriminator for FUDGE-Binary is a GPT-2 small model followed by multiple linear classification layers, with each one corresponding to an option for the label or length attribute. Each classification layer makes a binary prediction about whether the generation prefix satisfies the particular option of the attribute.

5 Experiments

We design experiments to test natural language CTG models' generalization capabilities, where the models need to generalize to (1) unseen commands (2) unseen attribute options (3) unseen combinations of attribute options. Additionally, we compare natural language CTG models with their categorical counterparts under the standard full-data setting to test whether the NL interface would degrade the model's performance.

5.1 Generalization to Unseen Commands

A key challenge introduced by the new interface is the diversity of natural language: commands with different surface forms can have the same underlying semantic. Thus we design a set of experiments to test natural language CTG models' ability to generalize to commands unseen during training. Specifically, we compare the effectiveness of our proposed PCFG with commands generated by fix-set templates, as adopted in previous works (Sanh et al., 2021; Wei et al., 2022; Mishra et al., 2022).

To create a setup similar to previous work, we hand-crafted 20 diverse templates for each dataset. This is already twice the number of templates used in Wei et al. (2022) and comparable to the number of seed templates in our PCFG. We denote models trained on this set of templates by "-T20" suffix. We also explore a stronger version of fix-set templates by doubling the number of templates, totaling 40 templates for each dataset, denoted by "-T40" suffix. We test the above models on 20 hand-crafted unseen templates that are different

[3] A publicly-available replication of GPT-3 obtained from https://huggingface.co/EleutherAI/gpt-neo-1.3B.

[4] Our code and data will be released in the public version of this manuscript.

Dataset	Method	*Control Accuracy*			*Text Quality*		*Diversity*
		Label ↑	Length ↑	Comp. ↑	G-PPL ↓	BLEU ↑	Ent. ↑
AG News	PrefixLM-NL-T20	.922	.522	.458	12.345	.865	11.412
	PrefixLM-NL-T40	.923	.496	.424	11.981	.863	11.405
	PrefixLM-NL-PCFG	**.933**	**.567**	**.505**	12.350	.868	11.381
	FUDGE-NL-T20	.936	.717	.603	11.677	.864	11.368
	FUDGE-NL-T40	.938	.759	.664	11.678	.864	11.355
	FUDGE-NL-PCFG	**.955**	**.936**	**.826**	12.174	.863	11.369
Yelp Review	PrefixLM-NL-T20	.389	.612	.177	10.523	.943	11.916
	PrefixLM-NL-T40	.398	.603	.216	10.309	.943	11.935
	PrefixLM-NL-PCFG	**.443**	**.721**	**.250**	10.251	.945	11.869
	FUDGE-NL-T20	.364	.531	.148	9.567	.936	12.155
	FUDGE-NL-T40	.538	.619	.249	9.986	.944	11.918
	FUDGE-NL-PCFG	**.687**	**.864**	**.462**	10.341	.941	11.836

Table 2: Results for experiment on PCFG effectiveness. Training NL CTG models with PCFG-generated commands greatly improves controllability on unseen commands, compared to models trained on fixed-set templates.

from both the PCFG and fixed-set templates, and compare results with our proposed PCFG-based models, denoted by "-PCFG" suffix.

The results in Table 2, show that when conditioning on unseen commands, both the PrefixLM-NL and FUDGE-NL models with PCFG have notably better controllability compared to fixed-set template models. The above experiments provide empirical evidence that **our PCFG can effectively improve the model's generalization ability on natural language variation within commands**.

5.2 Generalization to Unseen Attributes

CTG models with categorical attributes can only control a fixed set of attribute options. It is impossible for these models to control unseen attribute options without re-training due to architecture constraints (e.g., FUDGE trains a classifier with a fixed number of labels). In contrast, our proposed NL interface naturally allows CTG models to generalize to unseen options by embedding novel attributes into an NL command using a verbalizer phrase unseen during training, as long as the novel attributes could be described in natural language. In this section, we conduct experiments to test our PCFG-based natural language CTG models' capabilities to generalize control to unseen attribute options.

Experimental setup In this section, we control a single attribute (topic) for ease of presentation. Although it is possible to also experiment on the length attribute, they are similar in nature. For an attribute with n classes (e.g., 4 different topics), we create n zero-shot data splits and delete examples from one of the n classes (i.e. the zero-shot class) completely during training. We test on both the zero-shot and other seen classes separately and report the average result over all n splits. We conduct zero-shot experiments on the AG News dataset.

Adding extra data Since natural-language CTG models do not assume the attribute is from a fixed set of options, it is possible to train the model to control attributes by using extra data with different attribute options. This is another capability enabled by our NL interface, previously unavailable due to the fix-set assumption. We experiment training the models on the zero-shot AG News split along with similar datasets in the news domain, aiming to test whether the model can learn from extra data and generalize to a wider range of attribute options. We utilize three extra news topic classification datasets: News Popularity, News Category (Misra, 2022; Misra and Grover, 2021), and the Inshorts News dataset.[5] Topics that overlap with AG News are removed. We refer readers to Appendix A for more details. For these datasets, we use the same PCFG as AG News. When mixing multiple datasets during training, we follow Raffel et al. (2020) and use examples-proportional mixing to control the relative frequency of examples from each dataset. We set the artificial limit of each extra dataset to the size of the original AG News dataset.

The zero-shot results are shown in Table 3. Since the categorical interface does not allow unseen categories, we introduce a no-control baseline by fine-tuning the base LM with the same zero-shot data and producing generations from it directly without control. Both FUDGE-NL and PrefixLM-NL beat

[5]Obtained from Hugging Face Datasets and Kaggle.

		Control		*Text Quality*				*Diversity*	
		ACC. ↑		G-PPL ↓		BLEU ↑		ENT. ↑	
SETUP	METHOD	Z.S.	Reg.	Z.S.	Reg.	Z.S.	Reg.	Z.S.	Reg.
No Control Baseline	GPT-2-finetuned	.009	.343	11.050	11.062	.866	.867	9.745	9.735
Zero-shot data	PrefixLM-NL	**.222**	**.967**	14.797	11.556	.867	.860	9.736	9.726
	FUDGE-NL	.038	.927	21.604	11.497	.601	.863	9.359	9.748
	PrefixLM-NL-unb	**.204**	**.913**	12.980	11.387	.871	.862	9.738	9.737
	FUDGE-NL-unb	.203	.773	21.547	11.795	.623	.862	9.537	9.762
+Extra data	PrefixLM-NL	**.448**	**.960**	17.559	12.521	.868	.860	9.772	9.759
	FUDGE-NL	.071	.935	22.727	11.430	.782	.863	9.536	9.741
	PrefixLM-NL-unb	**.455**	**.928**	14.611	11.716	.867	.861	9.734	9.752
	FUDGE-NL-unb	.416	.784	24.898	11.933	.769	.864	9.587	9.748

Table 3: Results for zero-shot setting. Z.S. (zero-shot) denote metrics computed with the zero-shot class, REG. (regular) denote metrics computed with seen classes during training. The simple PrefixLM-NL approach outperforms FUDGE-NL. Adding extra data doubles the zero-shot accuracy.

		Compositional Accuracy			*Text Quality*		*Diversity*
DATASET	METHOD	TEST ↑	ORIG. ↑	DIFF. ↓	G-PPL ↓	BLEU ↑	ENT. ↑
AG News	PrefixLM-NL	**.593**	.612	**.019**	11.793	.861	10.293
	FUDGE-NL	.548	.914	.366	57.295	.677	10.140
Yelp Review	PrefixLM-NL	**.537**	.547	**.010**	13.831	.944	10.892
	FUDGE-NL	.046	.640	.551	19.335	.779	9.725

Table 4: Results for compositional setting. TEST denote accuracy for unseen attribute combinations, ORIG. denote accuracy in full-data setting, and DIFF. shows the difference. PrefixLM-NL suffers little performance loss when generalizing to unseen attribute combinations, but FUDGE-NL's performance substantially degrades.

this baseline.

We observe that the simple PrefixLM-NL approach outperforms FUDGE-NL by a large margin in both zero-shot data and zero-shot + extra data setting. Moreover, as measured by both perplexity and BLEU, PrefixLM has higher generation quality as well. While there is still a large gap between the zero-shot and non-zero-shot label accuracy, **the extra data approach managed to *double* the zero-shot accuracy in both NL models, showing the generalization potential of the natural language interface**. Qualitatively (shown in Table 8 to Table 11), we found that in cases where the output has the wrong topic, there are still signs that the generation is guided by the command. For example, when we zero-shot on the *world* topic, we obtain text about sports with multiple country names.

Backbone unblock experiment Due to the nature of the zero-shot experiment, we also block examples of the zero-shot class from the finetuning data of the backbone language model P_b. As a comparison, we try finetuning P_b with full data, while still blocking the zero-shot class from prefix or classifier training, which mimics the setting where only unlabeled data is available.

Results are shown in Table 3 as the "-unb" models. We observe a large performance boost for the FUDGE-NL model. This shows that extra unsupervised data is also helpful for control generalization.

5.3 Generalization to Unseen Attribute Combinations

In this section, we design experiments to test whether the models can generalize to unseen *combinations* of attributes to test their compositional generalization abilities. We describe our setup for AGNews below, which is similar to Yelp.

Following Lake and Baroni (2018), for each split, we select one of the topic classes (e.g., sports) as the non-compositional class, and for all training samples with this class, we do not include length in attributes or commands (i.e., the model never see combinations of `sports` and any length attribute in training). Note that the combinations of length attributes and other topics classes are kept (e.g., the model still sees combinations of `business` and `short length`). At test time, we set the topic to be the non-compositional class and randomly sample

Dataset	Method	*Control Accuracy*			*Text Quality*		*Diversity*
		Label ↑	Length ↑	Comp. ↑	G-PPL ↓	BLEU ↑	Ent. ↑
AG News	PrefixLM	.907	.559	.574	11.369	.862	11.325
	PrefixLM-NL	**.933**	**.677**	**.612**	12.126	.866	11.371
	FUDGE	.963	.962	.880	12.055	.862	11.286
	FUDGE-Binary	**.980**	.958	**.918**	12.617	.864	11.276
	FUDGE-NL	.965	**.972**	.914	12.197	.865	11.368
Yelp Review	PrefixLM	**.644**	**.949**	**.590**	10.406	.942	11.800
	PrefixLM-NL	.637	.919	.547	10.361	.943	11.828
	FUDGE	.620	.794	.564	10.628	.940	11.217
	FUDGE-Binary	**.871**	.942	**.805**	10.402	.943	11.727
	FUDGE-NL	.775	**.972**	.640	10.410	.941	11.802

Table 5: Results for full-data setting. NL model performance is on par with their categorical counterparts.

the length attribute to control. We run experiments across all n possible compositionality splits and report the averaged result.

Results are shown in Table 4, with qualitative examples available in Table 12 to Table 15. We focus on the accuracy gap between this compositionality setting and the full-data setting. PrefixLM-NL has little trouble generalizing to unseen attribute combinations as indicated by the small gap. However, FUDGE-NL performed poorly on generalizing to unseen attribute combinations. Not only did FUDGE-NL's compositional accuracy drop by a large margin, but it also produced low-quality text.

5.4 Full-data Setting

In the full-data setting, we train the models on all data of the AG News or Yelp review dataset, with the purpose to test whether the new NL interface would degrade the model's performance. This is the regular setup for existing works on CTG except that we aim to control two attributes simultaneously instead of one. The results for the full-data setting are shown in Table 5, with qualitative examples available in Table 6 and Table 7 in the appendix.

Performance comparison between the NL and categorical interface We notice that the generated text quality and diversity between different models are similar in the full-data setting. While PrefixLM-NL and its categorical variant PrefixLM have similar control accuracy on both datasets, FUDGE-NL consistently outperforms the original FUDGE setup. In either case, the performance of the NL variant is on par with its original model, suggesting our NL interface does not degrade CTG performance in the full-data setting. Somewhat surprisingly, FUDGE-Binary outperforms FUDGE-NL and the original FUDGE model, especially on the Yelp dataset where the classification is more difficult. The reason could be that the task of the binary classification is less noisy than the multiclass classification, which leads to stronger control.

Performance across model families Across two datasets, FUDGE-based models outperform PrefixLM models, with the exception that FUDGE does not beat (but is comparable to) PrefixLM on Yelp. This is largely consistent with previous results that discriminator-based CTG approaches can achieve higher controllability than conditional LMs (Yang and Klein, 2021, *inter alia*). However, as we show in the previous sections, its performance is inferior in the settings requiring NL generalization.

6 Conclusion

In this work, we propose a natural language interface for CTG, where we craft a PCFG to embed categorical attributes into natural language commands. We propose variants of existing CTG models that take commands as input. We design tailored experiments to test the natural language CTG model's generalization capabilities. We show that our PCFG-based command generation approach is effective for handling unseen commands compared to fix-set templates. Additionally, our proposed NL models can effectively generalize to unseen attributes, an ability newly enabled by the NL interface. Finally, we find the simple PrefixLM approach shows robust generalization ability with the NL interface and outperforms FUDGE-based models, demonstrating significant modeling challenges and potentials with this new interface. We hope our work could motivate further research into this challenging interface for CTG.

Limitations

In this section, we point out several limitations restricted by the scope of our work. While the PCFG we create has decent diversity and is guaranteed to be accurate in embedding attributes, they are still rule-based and could not cover all the variations in natural language.

The natural language interface brings modelling challenges. The CTG model is now required to first extract salient information from the command sentence, while in the original categorical interface they are provided directly.

In this work, we have focused our experiments on PrefixLM and FUDGE. While these approaches are representative, there are still other relevant models we did not test. For instance, guiding the generation of an LM with a smaller LM (Liu et al., 2021), or prompt-based CTG approaches such as Yang et al. (2022). It would also be interesting to test how other models perform under the NL interface.

Finally, while we experiment with controlling more than a single attribute in a single CTG model, in principle a NL command could be more complex and fine-grained. For example, it is possible to describe detailed semantic or syntactic constraints in a command sentence, and we leave those to future work.

Ethics Statement

We acknowledge controlled text generation is potentially capable of generating harmful outputs such as producing offensive languages or hate speech. However, it is also shown in previous work that controlled text generation techniques can achieve text detoxification if used properly (Dathathri et al., 2020; Krause et al., 2021). When changing the control interface from a categorical setting to natural language commands, we are giving the user a larger freedom of input. Thus, extra care should be taken when deploying natural-language controlled text generation models to the general public to avoid malicious user inputs.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jordan Clive, Kris Cao, and Marek Rei. 2022. Control prefixes for parameter-efficient text generation. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 363–382, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. 2022. Training compute-optimal large language models. *ArXiv*, abs/2203.15556.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*.

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

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. 2022. Diffusion-lm improves controllable text generation. *ArXiv*, abs/2205.14217.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Yuning Mao, Xiang Ren, Heng Ji, and Jiawei Han. 2020. Constrained abstractive summarization: Preserving factual consistency with constrained generation. *CoRR*, abs/2010.12723.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*.

Rishabh Misra. 2022. News category dataset. *arXiv preprint arXiv:2209.11429*.

Rishabh Misra and Jigyasa Grover. 2021. *Sculpting Data for ML: The first act of Machine Learning*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. 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)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

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

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 837–848, Dublin, Ireland. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang A. Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M SAIFUL BARI, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Rose Biderman, Leo Gao, T. G. Owe Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *ArXiv*, abs/2110.08207.

Timo Schick and Hinrich Schütze. 2021. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaden Smith, Mostofa Ali Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Anand Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie

Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *ArXiv*, abs/2201.11990.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam M. Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, Yaguang Li, Hongrae Lee, Huaixiu Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, I. A. Krivokon, Willard James Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Hartz Søraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Díaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravindran Rajakumar, Alena Butryna, Matthew Lamm, V. O. Kuzmina, Joseph Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *ArXiv*, abs/2201.08239.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2022. Tailor: A prompt-based approach to attribute-based controlled text generation. *ArXiv*, abs/2204.13362.

Dian Yu, Zhou Yu, and Kenji Sagae. 2021. Attribute alignment: Controlling text generation from pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2251–2268, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and William B. Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *NeurIPS*.

A Dataset Details

A.1 Main datasets

Yelp Review This is a dataset of user-written reviews for Yelp. It is a text classification dataset where the 5-sentiment labels are inferred from 1 to 5 stars given to the review. For each star, there are 130,000 training examples and 10,000 testing examples. In total, there are 650,000 training examples and 50,000 testing examples. We limit text length to 200 after tokenization. After this preprocessing step, there are 450,773 training and 34,620 testing examples, for a total of 485,393 examples. We sample a validation set from the train set with about the same size as the test set, and create a final dataset with 415,901/34,872/34,620 train/val/test examples.

The label attribute for Yelp Review is constructed from the 5 sentiment labels, which we verbalize as {very negative, negative, neutral, positive, very positive}. For the length attribute, we create 5 length classes {very short, short, medium-length, long, very long} with cut-offs 43,72,104,144 so that number of training examples in each length class is balanced. The dataset is obtained from https://huggingface.co/datasets/yelp_review_full.

AG News This is a news topic classification dataset with 4 topics {world, sports, business, science/tech}. The news text used is the title and description. For each topic, there are 30,000 training examples and 1,900 testing examples, for a total of 120,000 training and 7,600 testing examples. We limit text length to 256 after tokenization. After this pre-processing step, there are 119,955 training and 7,599 testing examples, for a total of 127,554 examples. We sample a validation set from the train set with about 10% of the original train set size, and create a final dataset with 107,959/11,996/7,599 train/val/test examples.

We use the topic labels as the label attribute, while adding alternative names for the labels. For the length attribute, we limit text length to 256. Because the text length in AG News is concentrated in a narrow range, we create 3 length classes {short, medium, long} with cut-offs 43 and 56 to make the number of training examples in each class balanced. The dataset is obtained from https://huggingface.co/datasets/ag_news.

A.2 Extra data

News Category The News Category dataset contains about 200K news headlines and short descriptions between 2012 and 2018 obtained from HuffPost. The advantage of this dataset is that it has a wide variety of topics, thus making the corresponding template very diverse. The list of topics and corresponding article counts is shown in Listing 1. We remove topics that has overlap with AG News: `THE WORLDPOST`, `WORLDPOST`, `WORLD NEWS`, `SPORTS`, `BUSINESS`, `SCIENCE`, `TECH`. The dataset is obtained from https://huggingface.co/datasets/Fraser/news-category-dataset.

News Popularity The News Popularity in Multiple Social Media Platforms dataset is a dataset of social media sharing data of news articles about `economy`, `microsoft`, `obama`, and `palestine`. We use the concatenation of the `headline` and `short_description` fields as the news text. The size of this dataset is around 93K. The dataset is obtained from https://huggingface.co/datasets/newspop.

Inshort News The Inshort News dataset is a dataset of news with topics `sports`, `politics`, `entertainment`, `world`, `automobile`, and `science`. We remove the topics that has overlap with AG News: `sports`, `world`, `science`. The filtered dataset contains about 5K examples. The dataset is obtained from https://www.kaggle.com/datasets/kishanyadav/inshort-news.

B Experiment Details

B.1 Training

On AG News, we use an Adam optimizer with a learning rate 0.00005 and train 10 epochs to train the PrefixLM models as well as FUDGE discriminators. On Yelp Review, we use an Adam optimizer with a learning rate of 0.0001 and train 5 epochs. We conduct all experiments on a single NVIDIA Tesla V100 GPU with 32GB memory. The training time of each model depends on the particular setup, but is within 24 hours for all models. The number of trainable parameters for the PrefixLM, PrefixLM-NL, and FUDGE-NL model is approximately 120M.

The number of trainable parameters for FUDGE and FUDGE-Binary is approximately 120M for each of label or length attribute model, and approximately 240M in total.

Listing 1: News Category dataset topics with corresponding number of examples.

```
POLITICS: 32739
WELLNESS: 17827
ENTERTAINMENT: 16058
TRAVEL: 9887
STYLE & BEAUTY: 9649
PARENTING: 8677
HEALTHY LIVING: 6694
QUEER VOICES: 6314
FOOD & DRINK: 6226
BUSINESS: 5937
COMEDY: 5175
SPORTS: 4884
BLACK VOICES: 4528
HOME & LIVING: 4195
PARENTS: 3955
THE WORLDPOST: 3664
WEDDINGS: 3651
WOMEN: 3490
IMPACT: 3459
DIVORCE: 3426
CRIME: 3405
MEDIA: 2815
WEIRD NEWS: 2670
GREEN: 2622
WORLDPOST: 2579
RELIGION: 2556
STYLE: 2254
SCIENCE: 2178
WORLD NEWS: 2177
TASTE: 2096
TECH: 2082
MONEY: 1707
ARTS: 1509
FIFTY: 1401
GOOD NEWS: 1398
ARTS & CULTURE: 1339
ENVIRONMENT: 1323
COLLEGE: 1144
LATINO VOICES: 1129
CULTURE & ARTS: 1030
EDUCATION: 1004
```

The FUDGE models have an extra backbone language model that is kept frozen during discriminator training. The size of this backbone language model is approximately 120M. Backbones are first fine-tuned on corresponding classification datasets with a learning rate of 0.0001 for 5 epochs.

B.2 Hyperparameter choice under different settings

We find that the experimental results are not particularly sensitive to training hyperparameters such as learning rate and batch size. At testing, the FUDGE conditioning strength hyperparameter λ does have a notable effect on control accuracy. We report results with λ that gives the highest control accuracy while maintaining text quality. For the FUDGE model family (FUDGE, FUDGE-Binary, FUDGE-NL), we set $\lambda = 14$ on the full-data and low-resource experiments, and $\lambda = 6$ on zero-shot experiments. On compositionality experiments, we set $\lambda = 6$ for AG News and $\lambda = 4$ for Yelp Review. We set a smaller λ for zero-shot and compositionality settings because a larger λ in these cases leads to a significant increase in repetition. Following FUDGE's original setup, we consider only the top 200 possible output tokens when modifying the LM logits for computational efficiency.

C Command PCFG Details

The full template for the AG News and Yelp Review datasets are available in Listing 2 and Listing 3. We briefly explain important elements of the custom PCFG syntax below:

- We first randomly sample a template in the `<templates>` section. These are templates with attribute slots which will be filled later. Besides attribute slots, there are other non-terminals in the template that corresponds to sentence segments. Rules for these elements are written in the `<variables>` sections.
- Rules in the `<variables>` sections are compressed PCFG where rules with the same LHS are grouped together in a single line. They constitute the verbalization of domain names, attribute names, as well as a variety of sentence segments to increase the diversity of the PCFG.
- To verbalize the label attribute, the `<label>` section contains the mapping from categorical class indices to verbalized class names. Since the mapping could be one-to-many, different verbalizations of the same attribute class is separated by a comma.
- To verbalize the length attribute, the `<length>` section contains length cut-off values with the corresponding verbalized length level names, having similar syntax with the `<label>` section. An example with tokenized length l will be treated as the longest length level such that the corresponding cut-off does not exceed l.

D Qualitative Examples

We show qualitative examples for different experimental settings in Table 6 to Table 15.

Listing 2: PCFG template for AG News

```
<variables>
[TEXT-CLASS]  AG news, AG news
[TEXT-FORM]   [TEXT-CLASS], [TEXT-CLASS], [TEXT-CLASS] article, piece of [TEXT-CLASS], [TEXT-CLASS]
    report, [TEXT-CLASS] item, AG newspaper article
[HEAD-FORM]   give me, generate, tell me about, show, show me, fetch me, output, I need, I want,
    need, I request, write
[TOPIC-NOUN]  topic, topic, theme, focus
[TOPIC-NOUNED] topic, topic, themed, focused, related
[TOPIC-PREP]  about, related to, concerning, regarding, pertinent to
[TOPIC-UPDATEWORD]   updated, informed
[TOPIC-SEG]   [TOPIC-PREP] [TOPIC], [TOPIC-PREP] [TOPIC], that is [TOPIC-PREP] [TOPIC], that is [
    TOPIC-PREP] [TOPIC], that can keep me [TOPIC-UPDATEWORD] with [TOPIC]
[TOPIC-BESEG] [TOPIC-PREP] [TOPIC], [TOPIC-PREP] [TOPIC], [TOPIC-PREP] [TOPIC], can keep me [TOPIC-
    UPDATEWORD] with [TOPIC]
[PLS]  please, ,
[COMMA-PLS]   / please, , # use '/' as comma (escaped)
[BEFORE-BE]   let it, make sure to, I want it to

<length>
43     short, concise, very short, pretty short, extremely short, extra short
56     medium-length, normal-length
256    long, lengthy, very long, pretty long, extremely long, extra long

<label> [TOPIC]
0      the world, the world, the globe, international matters
1      sports, sports, sporting events
2      business, business, commerce
3      science, science, technology, technology, tech

<templates>
# label and length
[HEAD-FORM] a [LENGTH] [TEXT-FORM] [TOPIC-SEG] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [TEXT-FORM] . [BEFORE-BE] be [LENGTH] and [TOPIC-BESEG] .
[PLS] [HEAD-FORM] a [TEXT-FORM] [TOPIC-SEG], and I need it to be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM] [TOPIC-SEG] , and [BEFORE-BE] be [LENGTH] [COMMA-PLS] .
[HEAD-FORM] a [TEXT-FORM] . I want the [TOPIC-NOUN] to be [TOPIC], and length to be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM] . I want the length to be [LENGTH], and [TOPIC-NOUN] to be [TOPIC] .
[HEAD-FORM] a [TEXT-FORM] . [BEFORE-BE] be not only [LENGTH] but also have a [TOPIC-NOUN] on [TOPIC]
    .
# label only
[HEAD-FORM] a [TOPIC] [TOPIC-NOUNED] [TEXT-FORM] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [TOPIC] [TOPIC-NOUNED] [TEXT-FORM] .
[HEAD-FORM] a [TEXT-FORM] [TOPIC-SEG] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [TEXT-FORM] [TOPIC-SEG] .
[PLS] [HEAD-FORM] a [TEXT-FORM] . Let it have a [TOPIC] [TOPIC-NOUN] .
[HEAD-FORM] a [TEXT-FORM] . Let it have a [TOPIC] [TOPIC-NOUN] [COMMA-PLS] .
[HEAD-FORM] a [TEXT-FORM] . I want the [TOPIC-NOUN] to be [TOPIC] .
# length only
[HEAD-FORM] a [LENGTH] [TEXT-FORM] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [TEXT-FORM] . [BEFORE-BE] be [LENGTH] .
[PLS] [HEAD-FORM] a [TEXT-FORM] , and I need it to be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM] , and [BEFORE-BE] be [LENGTH] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [TEXT-FORM] , and [BEFORE-BE] be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM]. I want the length to be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM]. I want the length to be [LENGTH] [COMMA-PLS] .
```

Listing 3: PCFG template for Yelp Review

```
<variables>
[TEXT-CLASS]  yelp review, yelp review, yelp comment
[TEXT-FORM]   [TEXT-CLASS], [TEXT-CLASS], [TEXT-CLASS] article, [TEXT-CLASS] passage, [TEXT-CLASS]
    paragraph, [TEXT-CLASS] piece, piece of [TEXT-CLASS], yelp review chapter, [TEXT-CLASS] item
[HEAD-FORM]   give me, generate, tell me about, show, show me, fetch me, output, I need, I want,
    need, I request, write
[SENT-NOUN]   tone, sentiment, attitude, mood
[SENT-PREP]   with, with, with, that has, / which has, of
[SENT-SEG]    [SENT-PREP] a [SENT] [SENT-NOUN]
[PLS]  please, ,
[COMMA-PLS]   / please, , # use '/' as comma (escaped)
[BEFORE-BE]   let it, make sure to, I want it to

<length>
43     very short, pretty short, extremely short, extra short
72     short, concise
104    medium-length, normal-length
144    long, lengthy
200    very long, pretty long, extremely long, extra long

<label> [SENT]
0      very negative, terrible, very bad, extremely negative
1      negative, bad
2      neutral, unopinionated
3      positive, good, promising
4      very positive, very good, excellent, splendid, extremely positive

<templates>
# label and length
[HEAD-FORM] a [LENGTH] [TEXT-FORM] [SENT-SEG] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [TEXT-FORM] . [BEFORE-BE] be [LENGTH] and having a [SENT] [SENT-NOUN] .
[PLS] [HEAD-FORM] a [TEXT-FORM] [SENT-SEG], and I need it to be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM] [SENT-SEG] , and [BEFORE-BE] be [LENGTH] [COMMA-PLS] .
[HEAD-FORM] a [TEXT-FORM] . I want the [SENT-NOUN] to be [SENT], and length to be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM] . I want the length to be [LENGTH], and [SENT-NOUN] to be [SENT] .
[HEAD-FORM] a [TEXT-FORM] . [BEFORE-BE] be not only [LENGTH] but also have a [SENT] [SENT-NOUN] .
# label only
[HEAD-FORM] a [SENT] [SENT-NOUN] [TEXT-FORM] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [SENT] [SENT-NOUN] [TEXT-FORM] .
[HEAD-FORM] a [TEXT-FORM] [SENT-SEG] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [TEXT-FORM] [SENT-SEG] .
[PLS] [HEAD-FORM] a [TEXT-FORM] . Let it have a [SENT] [SENT-NOUN] .
[HEAD-FORM] a [TEXT-FORM] . Let it have a [SENT] [SENT-NOUN] [COMMA-PLS] .
[HEAD-FORM] a [TEXT-FORM] . I want the [SENT-NOUN] to be [SENT] .
# length only
[HEAD-FORM] a [LENGTH] [TEXT-FORM] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [TEXT-FORM] . [BEFORE-BE] be [LENGTH] .
[PLS] [HEAD-FORM] a [TEXT-FORM] , and I need it to be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM] , and [BEFORE-BE] be [LENGTH] [COMMA-PLS] .
[PLS] [HEAD-FORM] a [TEXT-FORM] , and [BEFORE-BE] be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM]. I want the length to be [LENGTH] .
[HEAD-FORM] a [TEXT-FORM]. I want the length to be [LENGTH] [COMMA-PLS] .
```

PrefixLM-NL: Show me an AG news. I want the length to be concise, and topic to be the world. Two Indonesian women hostages released The two Indonesian women hostages have been released, two days before their scheduled arrival in the US under the terms of a new US policy in Southeast Asia.
FUDGE-NL: Generate an AG news article. Make sure to be not only pretty long but also have a focus on sporting events. No. 18 Utah Defeats Oregon, 71-60 (AP) AP - Cedric Benson scored 29 points, Jeri Dean added 15 and 18 points, and No. 18 Utah defeated Oregon 71-60 on Saturday night in an exhibition game that featured a balanced offense and a balanced defense.
PrefixLM: [WORLD][SHORT] Two Indonesian women hostages released Indonesia #39;s Foreign Ministry said Wednesday two female Indonesian women hostages were released Tuesday at dawn from an Al-Arabiya TV station, a day after it broadcast a videotape of their captivity.
FUDGE: [TECH][LONG] Macromedia #39;s #39;Blog #39; to Make New Web Sites Macromedia is setting out to make Web site-hosting Web sites, starting with blog, a key part of its strategy for making new sites more relevant and useful.
FUDGE-Binary: [BUSINESS][MEDIUM-LEN] Update 2: Oil Prices Drop Below $52 a Barrel Despite Hurricane Oil futures prices dropped below $52 on Friday, even though Hurricane Ivan, which was projected to hit the Gulf Coast late last week, was nonetheless expected to remain a drag on global oil supplies

Table 6: AG News generation in full-data setting. All models can provide sensible generations with descent quality.

PrefixLM-NL: Show me a yelp review passage. I want the length to be pretty short, and tone to be very negative. This place is a total joke. I called to place an order for delivery tonight and received a message with no answer.
FUDGE-NL: Generate a yelp review passage. Let it be long and having an extremely positive attitude. OMG I don't know where i got my hair cut. We started with the salon manager. I have to say I am a little disappointed on this. The place is nice, staff nice and the haircut was good. My stylist was very knowledgable. She did an excellent job with the color and cut. I was so impressed. I am going back here for my next cut. The staff is nice and very helpful. I will give it one more try before I leave. The price was very reasonable for it being in such a nice shopping center. I was surprised at the quality.
PrefixLM: [NEUTRAL][MEDIUM-LEN] Great for a quick meal but not really great. They do have a very good salad bar and the pizza is pretty good as well. If they had more seating and less of those annoying flies, I might have given them 4 stars. I am looking forward to trying more of their pizzas but they need more seating. I think I'll still go back, but not on a regular basis.
FUDGE: [NEGATIVE][SHORT] Disorganized and unorganized. No water in the shower and no soap dispenser in the bathroom. Very noisy when you come in during lunch hour. Very disappointed and will not go back.
FUDGE-Binary: [POSITIVE][SHORT] Great location! The pizza's are huge and the crust was thin. We had the sausage, pepperoni and sausage pizza and sausage was delicious. If you're looking for a good pie to eat late night, this is definitely the spot.

Table 7: Yelp Review generation in full-data setting.

Show me an AG news that is about business. China Stocks Set to Open Lower (AP) AP - China's consumer and investment markets are set to open relatively slackened this week due in part to lower oil prices, but stocks were slightly higher Monday #151; despite falling in recent weeks on the strength of the dollar and other key economic indicators.
Give me an AG news item that can keep me updated with international matters, please. China #39;s economy booming, but challenges persist China #39;s economy is booming and it is struggling to meet growing economic challenges, but it faces tough foreign competition quot;from within its own ranks quot;, China #39;s top official said Sunday.
Show a sports related AG news report. Update 1: Tennis: Masters Cup Semi Draws Three Aussies Aussies were forced to face three Australian players they will face in next month #39;s semi-finals, a spokesman said Tuesday.
Show an AG newspaper article. I want the topic to be technology. Google #39;s initial public offering The initial public offering of the Internet search engine is expected to be much higher than Google Inc #39;s initial public offering.

Table 8: Zero-shot generation of the PrefixLM-NL model. Zero-shot class generation is shown. Interestingly, the model seems to be generating overlapping topics between zero-shot and non-zero-shot classes, e.g., the generated business news talks about China's stock market, which the model probably learned from the world topic that is not blocked.

Give me a technology topic AG news. Microsoft to launch Xbox One console in India Microsoft is set to announce an Xbox One console in India on December 1. According to ET Now, the company will announce the launch on December 10. Microsoft announced the console on December 8. Its rival PlayStation 4 is also coming soon.
Show me an AG news item. Let it have a world topic, please. India plans to launch second space station A second crew to the International Space Station will start service next year as India completes a five-year mission to the moon and comes within one day of meeting the International Space Station crew.
Give me a sporting events themed AG news, please. Tennis: Oleg Tsunoda, Serena Williams Open in London (Reuters) Reuters - Two days before the third consecutive U.S. Open, tennis's biggest event, Nov. 8, will be staged on a hilltop overlooking London's
I request a commerce related AG news, please. U.S. Pounds Oil, Prices Surge in Asia (AP) AP - The United States pounds oil as the world's largest petroleum consumer enters a pivotal year in growth, pushing prices for gasoline and heating oil to new lows in Asia and boosting the U.S. energy industry.

Table 9: Zero-shot generation of the PrefixLM-NL model trained with extra data. Zero-shot class generation is shown. In the tech generation, the Microsoft topic is from one of the extra datasets.

Write an AG news item. I want the topic to be the globe. Sports industry executives to be legal firms CEO's offices on businesses, companies, companies not business clients, companies executives, business travelers.
Give me a sporting events related AG news. Fierce Fighting Roars Past a Home on the Wild Atlantic OAST LONDON - A series of strong Atlantic storms ripped through a British home on Tuesday, unleashing heavy rains, fierce winds and ferocious waves that washed away scores
Please generate a piece of AG news that is regarding commerce. Oracle Wins Contingency Bid For PeopleSoft (AP) AP - Oracle Corp.'s Oracle Corp. has won its contract for PeopleSoft Inc.'s business software division, Oracle Corp. said Friday.
Show an AG newspaper article. Let it have a technology theme, please. UN envoy urges Sudan resolution Sudanese authorities in Africa's top government urged Sudan to adopt a UN resolution on humanitarian crises in Africa in an effort to end fighting in the war-torn country.

Table 10: Zero-shot generation of the FUDGE-NL model. Zero-shot class generation is shown. In many cases the model ignores the command, and goes on to generate text related to seen topics.

Please generate a piece of AG news that is regarding the world. Paisley, Sharman hold India to 4 wickets for second innings (AFP) AFP - Shane Warne's unbeaten 141 against South Africa at the WACA was bowled out in the second and final innings as Australia held India to four wickets for a total of 141.
Generate an AG news. Let it have a sports theme. Beltre Leads Red Sox Past Yankees 7-4 (AP) AP - Mark Buehrle pitched seven shutout innings of two-hit ball, and Boston clinched its eighth straight AL East crown, 7-4 over the New York Yankees on Saturday.
Give me a commerce related AG news. Yahoo! Earnings Fall in First Quarter Yahoo! Inc.'s (YHOO.O: Quote, Profile, Research) quarterly earnings fell sharply, as the world #39;s largest Internet advertising and search company reported narrower
Please show me an AG news. Let it have a technology focus. Tech giants IBM profits rise tech giants IBM and Hewlett-Packard reported big gains yesterday as their technology businesses grew more profitable in August. Shares of Hewlett-Packard jumped as much as 2.

Table 11: Zero-shot generation of the FUDGE-NL model trained with extra data. Zero-shot class generation is shown. Signs of the model being influenced by the command is evident from time to time, e.g., the world topic generation, although is actually sports-topic, contains many country names.

Generate an AG news. Make sure to be long and about international matters. US Forces Storm School in Mosul, 40 Dead MOSUL, Iraq (Reuters) - U.S. tanks, aircraft and artillery rumbled through a Mosul rebel-held school on Saturday, 40 days after a suicide car bomb blast killed at least 39 people outside a police academy in a town near Baghdad.
Need an AG news. I want it to be extra short and can keep me updated with sports. Soccer: England win draw at Moldova England won their opening game against Moldova after defeating Poland 1-0 in a friendly draw at Frno, the Football Association said Thursday.
I want an AG news report. I want it to be extremely short and can keep me informed with business. Oil Prices Surge to Record Highs on Ivan Crushing $52 Oil prices jumped to multi-week highs yesterday on fears that Hurricane Ivan would hit the Southeast US Gulf Coast early next week.
Give me an extra long AG news article related to science. Microsoft to Release Longhorn as Open Beta Microsoft announced it would release Longhorn in 2006, with features such as a new file system, new system management, and support for 64-bit extensions in addition to standard 64-bit capabilities.

Table 12: Compositional generation of the PrefixLM-NL model on AG News. Non-compositional class generation is shown. The model generate fluent, on topic text without much performance degradation from full-data setup.

Generate an AG news article. I want the focus to be international matters, and length to be pretty short. Italy arrests Italian businessman Italy yesterday arrested a businessman and a German national working for a major French and British industrial group and two others for connections to a major European political party scandal in Italy two years ago.
Give me a piece of AG news. I want the focus to be sporting events, and length to be long. NFL Wrap: Brees, Brees Lead the Giants to the Playoffs Jamal Lewis ran for 147 yards and two touchdowns Saturday night to lead the Giants to playoff berth games Sunday at Philadelphia and Sunday at Atlanta and clinch Sunday #39;s NFC title ...
Need an AG news article. Make sure to be not only medium-length but also have a focus on commerce. Oil prices rebound sharply from record highs <b&gt...</b&gt VIYUYYashko men took no charge as they trawl past what lookto evidence indicate Since arriving earlier earlier early today many children wandered past makeshift shelters and makeshift stalls. At least two dozen workers - many of them women - entered the battered warehouse, but as...
Generate an AG news article. I want the focus to be technology, and length to be pretty short. Computer security chief steps down Computer security chief William Donaldson steps down today, the government said today. Mr Donaldson was leaving his department in his final days of the career.

Table 13: Compositional generation of the FUDGE-NL model on AG News. Non-compositional class generation is shown. The text quality is notably lower than both PrefixLM's compositional generations (Table 12) as well as FUDGE-NL's generation in full-data setting (Table 6).

Write an extremely short yelp review passage of a very negative tone. This place is a joke!!!!!! I called to place an order for delivery tonight and they quoted me $50! And the food wasn't even good at all!!!
Write an extremely short yelp review passage of a negative tone. This place is a total joke. I wish I had just gone to another place before coming here. The food was ok (but not really worth wasting $40 on a meal) The service suck!!!
Generate a yelp review. Let it be concise and having an unopinionated attitude. Not bad, not great (except for the wait for service and for the food being cold). However, the place is really small and is always crowded. The food is good (although a bit pricey) and the ambiance is relaxing. But again not great.\nThe servers are nice and accommodating.
I need a yelp review item. I want the length to be extra short, and tone to be promising Very happy with this hotel! Good deals, clean room and great service. The staff was very helpful. Will stay here again.
I need a yelp review passage. I want the mood to be very good, and length to be very long I am a HUGE fan of this place! This is the type of place that I go to when I'm in the area! The food is fantastic! I have always had great service here. I usually go at dinner time on weekends but have yet to be disappointed. I love their wings! They are not your average wings. They come covered with a wonderful sauce and just a hint of garlic! They are always hot. I am a big guy though, so I tend to eat wings on the go. \n\nTheir pizza is amazing as well. I love that you can order a large size with no toppings, which is very convenient for me. I also love their ranch dressing, which is amazing as well. \n\nI always look forward to coming back to this place for some wings!

Table 14: Compositional generation of the PrefixLM-NL model on Yelp Review. Non-compositional class generation is shown. Similar to scenario in AG News (Table 12), the model can generalize to unseen attributes combinations without much performance degradations.

Show a yelp review. Let it be not only lengthy but also have an extremely negative tone. Love this place. Great coffee and delicious sandwiches, wraps, and salads. The service is wonderful and everyone is helpful and friendly.
Write a yelp comment that has a bad sentiment, and make sure to be short. I really wanted to love this place. I love the concept. The food. Unfortunately, I was not impressed.
I need a yelp review paragraph. Make sure to be not only short but also have a neutral attitude. It was ok. Great tasting bread with your sandwiches and cookies on your cookie crust to make it look like a fancy bakery with your cookies.... No wow... Really!?!?!?..... Never ever ever ever ever when ever I have been. Kind it kinda of defeats thinking about this at all. Service goes off without any notice.
I request a yelp review piece. I want the length to be extra long, and tone to be positive. So much for the good reviews. My sister-in-law was going to give this place a try, but it seems a little overrated.\n\nI came for lunch with her, and ordered a half salad, soup, and sandwich combo. My salad came with the soup, but only half of it was actually soup! The soup was alright, but it came with a small piece of lettuce on top. The sandwich was decent.\n\nThe sandwich was nothing spectacular. I've had better for half the price!
I want a piece of yelp comment that has a very good mood, and make sure to be pretty long. Always a great place. Food & service always great & prices are reasonable especially for the quality & quantity they give you. Food comes out hot. My kids eat there every time & are always happy with their meals. Prices have always been very reasonable for the quality & quantity they give you. Owner is the man, & he is the reason I come back to this place, & I hope he is getting his act together soon! Give it a try & please try them out for yourself!! You will leave happy & full!! :) Thanks Chef! Keep hustling for quality & quality food! Happy eating! Thanks Chef :) :)) :) Enjoy! :-) :) See ya! :) ;) :) :) :) :) :) :) ;) ;)

Table 15: Compositional generation of the FUDGE-NL model on Yelp Review. Non-compositional class generation is shown. Text quality is notably low, with the model generates repetitive phrases or emoji from time to time.