Critic-Guided Decoding for Controlled Text Generation

Minbeom Kim[1*] **Hwanhee Lee**[2] **Kang Min Yoo**[3,4,5]
Joonsuk Park[3,4,6] **Hwaran Lee**[3,4†] **Kyomin Jung**[1†]
[1]Seoul National University [2]Chung-Ang University [3]NAVER AI Lab
[4]NAVER Cloud [5]AIIS [6]University of Richmond
{minbeomkim, kjung}@snu.ac.kr, hwanheelee@cau.ac.kr
{kangmin.yoo, hwaran.lee}@navercorp.com, park@joonsuk.org

Abstract

Steering language generation towards objectives or away from undesired content has been a long-standing goal in utilizing language models (LM). Recent work has demonstrated reinforcement learning and weighted decoding as effective approaches to achieve a higher level of language control and quality with pros and cons. In this work, we propose a novel critic decoding method for controlled language generation (CriticControl) that combines the strengths of reinforcement learning and weighted decoding. Specifically, we adopt the actor-critic framework and train an LM-steering critic from reward models. Similar to weighted decoding, our method freezes the language model and manipulates the output token distribution using a critic to improve training efficiency and stability. Evaluation of our method on three controlled generation tasks, topic control, sentiment control, and detoxification, shows that our approach generates more coherent and well-controlled texts than previous methods. In addition, CriticControl demonstrates superior generalization ability in zero-shot settings. Human evaluation studies also corroborate our findings.

1 Introduction

With recent advances in large language models (LMs), generating natural-sounding text has become feasible (Radford et al., 2019; Brown et al., 2020; Kim et al., 2021). However, such text can still be undesirable; for instance, it may be off-topic or biased and otherwise offensive, reflecting the harms in the real-world data (Keskar et al., 2019; Liu et al., 2021; Gehman et al., 2020; Hosseini et al., 2017; Krause et al., 2020; Lin and Riedl, 2021; Qian et al., 2022; Meng et al., 2022). To address this issue, several controlled text generation methods have been proposed (Lu et al., 2022; Yang and Klein, 2021; Dathathri et al., 2019).

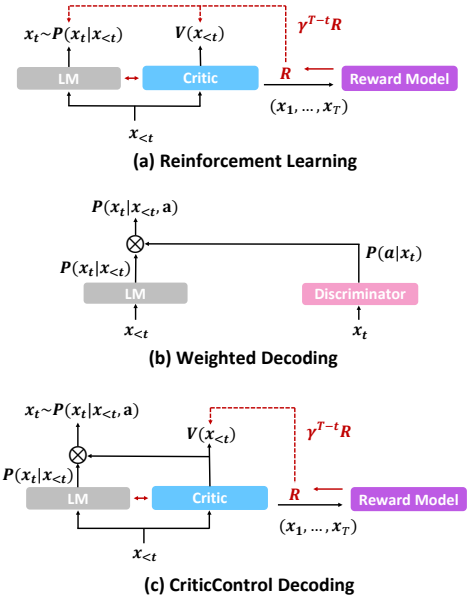

Figure 1: Overview of Controlled Text Generation Approaches. Note, (a) can generate fluent text through sequential decision-making, (b) allows effective and efficient control using a 'plug-and-play' discriminator, and (c) combines both strengths.

As shown in Figure 1, two major categories of approaches to controlled text generation currently exist: (a) reinforcement learning (RL) and (b) weighted decoding. In the RL approaches, generating a word at each time step is formulated as sequential decision-making leveraging an LM's probability distribution over words (Wu et al., 2016; Paulus et al., 2017). For example, in the widely used Actor-Critic framework, the *actor*—a pretrained LM—predicts the next word, and the *critic*—a value network—evaluates the state, i.e., text generated thus far (Stiennon et al., 2020; Wu et al., 2021). Unfortunately, LM fine-tuning through RL often suffers from noisy gradient estimation (Greensmith et al., 2004), which may lead to unstable training, and eventually mode collapse (Upadhyay et al., 2022). Even when fine-tuned correctly, optimizing an entire LM for each target attribute, e.g., detoxification, is computationally expensive and memory

[*] Work done during an internship at NAVER AI Lab.
[†] Co-corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2023, pages 4598–4612
July 9-14, 2023

inefficient for real-world applications (Guo et al., 2021).

In the weighted decoding approaches, the underlying LM is kept frozen, and only the final output probability distribution is adjusted (Holtzman et al., 2018; Dathathri et al., 2019; Kumar et al., 2021). More specifically, to condition an LM's output $p(x)$ on a target attribute, an approximation of Bayesian decomposition $p(x|a) \propto p(x)p(a|x)$ is computed with $p(a|x)$ from an external discriminator (Yang and Klein, 2021; Krause et al., 2020). Note, these approaches involve a frozen LM with an independent discriminator for each target attribute; In comparison to the RL approaches, this plug-and-play structure allows more efficient training and memory use (Gu et al., 2022), but degrades the text quality in terms of fluency, diversity, etc. (Lu et al., 2022).

In this paper, we propose a novel controlled text generation algorithm, *Critic-Guided Decoding for Controlled Text Generation (CriticControl)*, that re-weight the word distribution from an LM with predicted state-values from the critic network of RL. As a result, CriticControl raises the likelihood of words that increases the value of the next state over the current state while lowering that of others. Since the critic is trained with a frozen LM, the training is stable and efficient as in weighted decoding. Moreover, different critics can be used in a plug-and-play manner depending on targeted attributes or reward models. In other words, CriticControl combines the strengths of both the RL and the weighted decoding approaches.[1]

We demonstrate the efficacy of CriticControl through experiments on three controlled text generation tasks: topic, sentiment, and toxicity control. For all tasks, we find that CriticControl consistently outperforms previous methods in terms of control success, fluency, and diversity. Also, CriticControl exhibits strong zero-shot controllability on unseen topics. Finally, CriticControl is compatible with widely used sampling methods like top-k and top-p sampling (Holtzman et al., 2019; Fan et al., 2018) for improved text quality.

2 Related Works

Reinforcement Learning RL and the adversarial training formulation were first proposed in the context of language generation as an auxiliary algorithm to mitigate exposure bias in the teacher-forcing training of sequences (Ranzato et al., 2015; Wu et al., 2016; Hu et al., 2017). The main motivation is to incorporate readily-available sequence-level reward signals into training, such as BLEU or ROUGE (Paulus et al., 2017). The success of utilizing RL has been observed in a wide range of tasks, including summarization (Paulus et al., 2017; Wu and Hu, 2018; Stiennon et al., 2020; Ziegler et al., 2019), dialog modeling (Li et al., 2016; Yi et al., 2019; Jang et al., 2021; Upadhyay et al., 2022), neural machine translation (Wu and Hu, 2018; Nguyen et al., 2017), and style transfer (Gong et al., 2019; Ziegler et al., 2019). Furthermore, RL has allowed models to capture high-level human feedback (Paulus et al., 2017; Stiennon et al., 2020; Sharma et al., 2021; Ramamurthy et al., 2022), which is out of the current work's scope, however. Various RL approaches have been explored so far, such as REINFORCE (Sutton et al., 1999; Ranzato et al., 2015; Wu et al., 2016; Sharma et al., 2021; Upadhyay et al., 2022), the actor-critic framework (Bahdanau et al., 2016; Nguyen et al., 2017; Jang et al., 2021), and PPO (Schulman et al., 2017; Nakano et al., 2021; Snell et al., 2022). More recently, Critic-guided methods have been studied to avoid the risk of RL's policy learning collapse. GPT-Critic (Jang et al., 2021) generates critic-guided texts for enriching task-oriented dialogue datasets. ILQL (Snell et al., 2022)'s critics guide supervised models to mimic the behavior of reinforced models. Our work is the first to incorporate weighted decoding for sequence-level reinforcement learning.

Weighted Decoding Freezing the language model and controlling the output probability distribution to suit the purpose is being actively researched (Holtzman et al., 2018; Ghazvininejad et al., 2017; Keskar et al., 2019; Sudhakar et al., 2019). This mainstream approach can be used in various scenarios, such as forcing the model to generate text that conforms to certain stylistic or content-based constraints or mitigating aggressive and toxic expressions (Kumar et al., 2021; Gu et al., 2022; Mireshghallah et al., 2022). The PPLM (Dathathri et al., 2019) generates texts by plugging a steering layer into the top of language models $p(x)$. Then, the gradient from $p(a|x)$ updates iteratively the last hidden representation to desired attributes. It only needs a few layers per each attribute but requires a lot of iterative computations; FUDGE (Yang and Klein, 2021) economizes $p(x|a)$ by bayesian decomposition $p(x|a) \propto p(x)p(a|x)$

[1]Code will be available at https://github.com/minbeomkim/CriticControl.

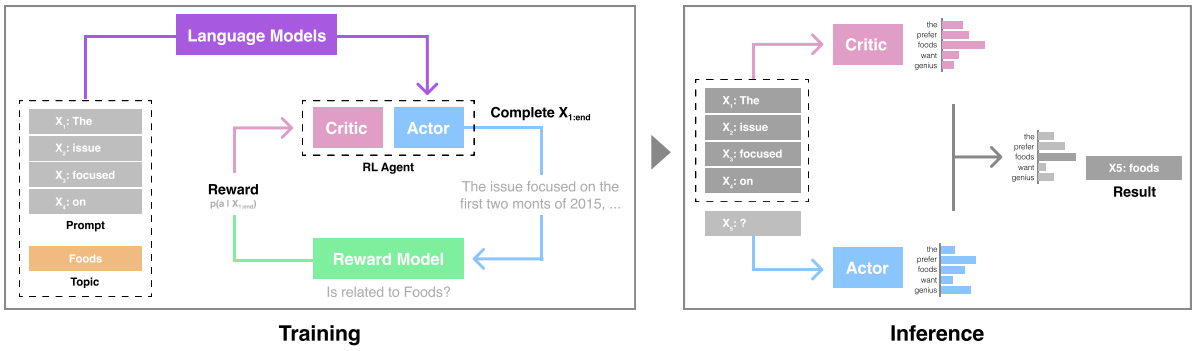

Figure 2: Overall flow of CriticControl. Language model completes the sentence in response to the prompt 'The issue focused on', and the reward model evaluates the output by judging the relevance of the generated text to the specified topic. During the inference, critic modifies the output distribution of the language model to ensure that the generated text is appropriately related to the topic when choosing the next token.

with $p(a|x)$ computing classifier instead of gradient update methods. For better linguistic quality, GeDi (Krause et al., 2020) and DExperts (Liu et al., 2021) take generative discriminator approaches with two LMs conflicting polarly on the desired attribute(e.g., positive vs. negative). They interpret $p(a|x)$ as the degree of disagreement between two conflicting LMs. In contrast, CriticControl is advantageous in taking a generative manner with only a steering layer, like PPLM.

3 CriticControl

CriticControl is a controlled text generation framework that consists of a frozen pre-trained language model (*Actor network*) and an extra network for the state-value prediction (*Critic network*). Given a part of the sentence, the actor tries to complete the attribute-aware continuations with the critic's support.

As an RL formulation, at each step t, states are given as a set of tokens $x_{1:t-1} = \{x_1, ..., x_{t-1}\}$ and the attribute a. The policy π_θ of the actor samples $x_t \in \mathcal{V}$ from the next tokens probability of $\pi_\theta(x_t|x_{t-1})$ as

$$P_\pi(x_t|x_{1:t-1}) = \text{softmax}\left(\frac{y_t}{T}\right), \qquad (1)$$

which output logits y_t for words in dictionary $\mathcal{V}$ with temperature T to experience diverse trajectories. To train the policy network (i.e., LM), we use widely used optimization methods (Sutton et al., 1999) that take policy loss as $\nabla_\theta J(\theta) = \mathbb{E}_\pi[\sum_{t=1}^{end} A_t \nabla_\theta \ln \pi_\theta(x_t|x_{1:t-1})]$. Note that A_t is the advantages function that measures how much the choice x_t is better than the critic predicted, and we train the critic to minimize A_t. In our algorithm, we freeze the actor model and only train the critic for flexible control, as described in the following section.

3.1 CriticControl Training

Different from the previous supervised training, we design a simple text generation framework via reinforcement critic learning. As shown in Figure 2, when the actor completes a given prompt, we let the reward model evaluate how well the completed sentence correlates with the attribute a. Then we give critic a reward $r_{end} = P_\pi(a|x_{1:end})$ to calculate the temporal difference (TD) error $\delta_t = r_t + \gamma V_\pi(x_{1:t+1}) - V_\pi(x_{1:t})$, consisting of A_t. This TD error generalizes $V_\pi(x_{1:end}) = P_\pi(a|x_{1:end})$ to $V_\pi(x_{1:t}) = P_\pi(a|x_{1:t})$ (Sutton and Barto, 2018). Using the evaluation result r_{end} and the text generation history $x_{1:end}$, CriticControl minimizes the generalized advantage estimation loss (Schulman et al., 2015) to train critic network as follows:

$$\mathcal{L}_{critic} = \sum_{t=1}^{end}\Big(\sum_{i=0}^{end-t} (\gamma\lambda)^i \delta_{t+i}\Big)^2, \qquad (2)$$

where γ is the reward discount factor and λ is the re-weighted averaging factor. We backpropagate this loss to align the critic's prediction $V_\pi(x_t)$ with unbiased empirical future returns to train the critic network. For the exploration in the training phase, we adopt a highly stochastic actor strategy. Specifically, an actor generates diverse outputs from language models by using high temperatures $T > 1$ as in Equation 1, which increases the entropy of output distributions for logits y_t over a vocab of tokens $\mathcal{V}$. And this leads to diverse text generation, and the critic can experience more diverse samples in

each episode. By repeating this simulation, we expect the critic to learn which decisions of language models will lead to a promising future.

3.2 Text Generation with CriticControl

Generating human-like text often requires stochastic decoding strategies. They truncate the unreliable long-tail on the probability distribution for sampling only on realistic token candidates (Holtzman et al., 2019; Fan et al., 2018). However, adjusting all probabilities in the vocabulary is very computationally inefficient. Therefore, to achieve both stochasticity and computational efficiency simultaneously, CriticControl steers the subset of vocabulary $\mathcal{V}' \subset \mathcal{V}$, consisting of top-$k$ probability tokens from frozen language models with exact probability re-weighting

$$\begin{aligned} P_\pi(x_t|x_{<t}, a) &= \frac{P_\pi(x_t, x_{<t}, a)}{P_\pi(x_{<t}, a)} \\ &= \frac{P_\pi(a|x_{\leq t})}{P_\pi(a|x_{<t})} P_\pi(x_t|x_{<t}), \end{aligned} \tag{3}$$

to align the probability scale of adjusted words $x \in \mathcal{V}'$ and non-adjusted words $x \notin \mathcal{V}'$. After adjusting the distribution, CriticControl can combine beam search and various sampling methods. In experiments, CriticControl adopts both top-k sampling and nucleus sampling by adjusting the top-10 word probabilities of $\mathcal{V}' \subset \mathcal{V}$.

4 Experiments

To evaluate the effectiveness of CriticControl, we conduct experiments on a variety of controlled text generation tasks, including topic control, sentiment control, and detoxification.

4.1 Topic Control

We conduct experiments on topic control tasks to generate topic-related text, starting with a prompt consisting of natural plain text independent of any topic. When implementing our framework, we use *BART-large-MNLI* (Lewis et al., 2019)[2] as a reward model that computes the relevance between the texts and the topics. During the training, the critic is randomly given 1 of 7 topics (computers, space, military, legal, politics, science, and religion) (Dathathri et al., 2019) and learns to predict the semantic relevance between given topics and generated texts. For the baselines, we report the results of pure *GPT-2-medium* (Radford et al., 2019), WDEC (Yang and Klein, 2021), PPLM, and FUDGE. Additional implementation details can be found in Appendix A.1.

4.1.1 Metrics and Evaluation

We evaluate the quality of the generated text in terms of success in controllability, fluency, and diversity. All baselines generate 80 tokens on 20 prompts $\times$ 7 topics = 140 comparisons. Previous works (Dathathri et al., 2019; Yang and Klein, 2021) measure controllability as the usage rate of their predefined topic-related words used for both training and evaluation. In contrast, our approach performs more general optimization and does not use any predefined topic-related words. Hence, we measure this 'success on control' with human evaluation. Annotations answer the question, 'Is the text relevant to a given topic?' for all generated texts. Additionally, we also run the human evaluation of topic control success on the unseen topic (i.e., zero-shot setting), not directly related to training topics. We measure the fluency using two metrics: perplexity, which is calculated using the *GPT-2-XL* language model, and grammaticality, which is determined using the *Roberta-based CoLA* model (Warstadt et al., 2019) [3]. Finally, we measure diversity (Li et al., 2015) using distinct n-grams normalized by text length, reporting distinct unigrams, bigrams, and trigrams as Dist-1, Dist-2, and Dist-3 scores, respectively.

Results As shown in Table 1, CriticControl significantly outperforms other baselines on topic controllability, fluency, and diversity. CriticControl achieves a superior control success rate, and this tendency is proportional to the size of the GPT-2 model. And even small-sized CriticControl beats the FUDGE in terms of topic controllability. Furthermore, CriticControl is able to well preserve the linguistic characteristics of the baseline GPT-2 as verified by the lower perplexity and the higher grammar scores. We argue that the sequential decision-making approach of CriticControl helps the system preserve the linguistic characteristics of the original GPT-2 during the training process. Additionally, CriticControl generates texts without any pre-defined "bag of words" about topics, unlike WDEC, PPLM, and FUDGE. This property makes the system freely choose the topic-related words and

[2] https://huggingface.co/facebook/bart-large-mnli

[3] https://huggingface.co/textattack/roberta-base-CoLA

Model	Success	Fluency		Diversity		
	On-Topic	Perplexity ↓	Grammar	Dist-1	Dist-2	Dist-3
GPT-2-medium (Radford et al., 2019)	0.16	**14.06**	0.74	0.29	0.70	0.88
WDEC (Yang and Klein, 2021)	0.49	67.53	0.59	0.16	0.42	0.85
PPLM (Dathathri et al., 2019)	0.45	62.66	0.78	0.35	**0.78**	**0.92**
FUDGE (Yang and Klein, 2021)	0.78	69.08	0.79	0.34	0.75	0.91
CriticControl	**0.89**	17.19	**0.83**	**0.49**	0.76	0.90
CriticControl - small	0.85	16.88	0.83	0.47	0.73	0.89
CriticControl - large	0.92	17.58	0.84	0.51	0.77	0.91
CriticControl - XL	**0.94**	17.69	0.83	0.51	0.77	0.91
CriticControl - Zero shot	**0.73**	17.55	0.85	0.49	0.76	0.90

Table 1: Evaluation results on topic control experiments using GPT-2. Success is the human evaluation results of 'being on the topic'. The other metrics are automatic evaluation results about text quality. The first five rows show comparisons over baselines steering freezed GPT-2-medium. Baseline results are adopted from FUDGE (Yang and Klein, 2021). In the second, CriticControl-[size] indicates the size of the freezed GPT-2. The last is an experiment on the generalization ability of CriticControl. Other than the above settings, this experiment uses entirely new topics to verify how robust control is possible, even on unseen topics.

Figure 3: Human preference test results for topic control. This experiment compares human preferences between two generations under the same prompts and topics, CriticControl vs. GPT-2-medium and CriticControl vs. FUDGE.

results in a higher Dist-1 score. These experimental results verify the effectiveness of this reward-driven controllable text generation system as sequential decision-making.

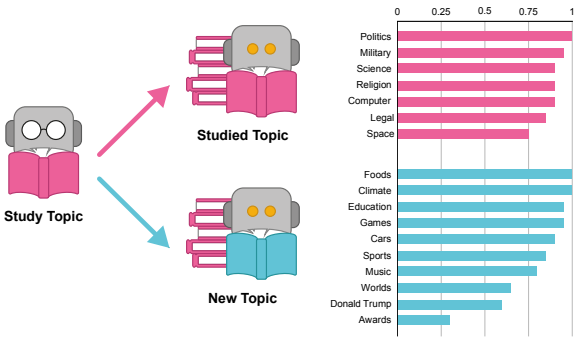

Figure 4: The CriticControl success rate per topic. It is examined for both the topic used for training and a new, previously unseen topic. The results indicate consistent performance on zero-shot topics.

4.1.2 Generalization on unseen Topics

Since the critic learns to predict semantic relevance between the given random topic and texts to generate, CriticControl is also able to generalize toward unseen topics other than seven training domains. In zero-shot results of table 1, CriticControl shows a high topic control success rate on new topics. As shown in Figure 4, CriticControl is even able to control zero-shot topics such as '*food, cars, sports*, and *music*' that are not seen during training. These results indicate that the usage of the general reward model makes it possible to evaluate the semantic relevance of unseen topic codes to the current texts. For example, in Table 3, CriticControl generates texts about Paul McCartney and the FDA from unseen topics music and food respectively, without using pre-defined dictionaries. These results prove that CriticControl obtains generalization ability by taking universal reward models, and this free setting of the reward model will enable various promising future research.

4.1.3 Human Preference Tests

There is a limitation to solely relying on automated evaluation for measuring text quality. Therefore, we run preference tests on CriticControl against GPT-2 and FUDGE to validate that the text quality of CriticControl outperforms previous baselines while being on topics. For the preference test, we hire three annotators for each comparison pair and ask two questions *Success Rate: 1) Which sentence is more related to the given topic?* and *Fluency: 2) Which sentence is more fluent?*.

Model	Success	Fluency		Diversity		
	Positiveness	Perplexity ↓	Grammar	Dist-1	Dist-2	Dist-3
GPT-2-medium (Radford et al., 2019)	0.57	**11.91**	0.78	0.25	0.63	0.78
PPLM (Dathathri et al., 2019)	0.60	142.11	0.73	0.22	0.61	0.72
CC-LM (Krause et al., 2020)	0.76	15.79	0.72	0.28	0.70	0.82
GeDi (Krause et al., 2020)	0.84	38.94	0.76	0.27	0.77	0.89
CriticControl	**0.90**	12.97	**0.87**	**0.31**	**0.84**	**0.92**
PPO	0.94	13.43	0.84	0.32	0.86	0.93
PPO - CriticControl	**0.99**	13.44	0.80	0.32	0.85	0.93

Table 2: Evaluation results on sentiment control language generation using GPT-2. The first is about comparison over steering freezed GPT-2-medium with each guided decoding method. The last is for verifying the control capability to improve even reinforced language models. This experiment compares GPT-2-medium finetuned on PPO and those PPO with CriticControl.

Figure 5: Human preference test results for sentiment control. This experiment draws comparisons on human preferences between two continuations after the same prompts, CriticControl vs. GPT-2 and CriticControl vs. GeDi.

Music *Emphasised are* the words "instrument" and "instrumentals" in the title. The song is a cover of the song "I'm a Man" by the band *The Beatles*. "I'm a man" is a reference to the song "Man of the World" by the British band The Beatles, which was written by *John Lennon and Paul McCartney*.
Foods *The issue focused on* the use of the term "organic" in the food industry. This issue focused on a new USDA regulation that requires food companies to label their products as "organic" if they meet certain criteria. The regulation was passed in 2010, but the *Food and Drug Administration (FDA)* has yet to issue a final rule.

Table 3: The zero-shot topic control examples on given topic-prompt pairs {*Music*, *Emphasised are*} and {*Foods*, *The issue focused on*}. CriticControl generates words like '*Beatles*' and '*USDA regulation*' that are less likely to observe in the training set.

Results In the preference test, CriticControl also outperforms baselines on both topic control success rate and fluency, as shown in Figure 3. In the fluency comparison with GPT-2, CriticControl also wins, resulting from generating sentences within the area guaranteed by the reward model. Both automatic and human evaluation results indicate that CriticControl achieves a state-of-the-art topic control ability while preserving the original GPT-2's text quality.

4.2 Sentiment Control

Next, we explore CriltticControl's ability to steer text generation toward a specific sentiment. This sentiment control task aims to steer the model to complete positive movie reviews with any emotional prompt. We adopt the IMDB movie review dataset, containing highly polar (positive or negative) reviews of 2.5K for training and 2.5K for testing. We use the 8 starting tokens for each sentence to make prompts and generate 25 continuations for each prompt using all of the baseline systems. In this task, we additionally test 'Could CriticControl enhance the reinforcement-learned LMs to achieve goals more appropriately?'. Additional implementation details are in Appendix A.2.

4.2.1 Metrics and Evaluation

We evaluate total $2.5K$ generations of each baseline in terms of positiveness, fluency, and diversity. We define the positiveness as the percentage of generations classified to 'positive' from *distilBERT* classifier finetuned on the IMDB dataset, used as our reward model. In the same way as the topic control task, we measure fluency through perplexity and grammatically, using Dist-n for diversity metrics.

Results As shown in Table 2, CriticControl significantly outperforms the other baselines on the success rate of sentiment control, fluency, and diversity metrics. Also, we observe that generative controllers such as GeDi and CriticControl demon-

Model	Success	Fluency		Diversity		
	Toxic prob ↓	Perplexity ↓	Grammar	Dist-1	Dist-2	Dist-3
GPT-2-large (Radford et al., 2019)	0.520	**11.31**	**0.84**	0.58	0.85	0.85
PPLM (Dathathri et al., 2019)	0.518	32.58	0.75	0.58	**0.86**	0.86
DAPT (Gururangan et al., 2020)	0.360	31.21	0.71	0.57	0.84	0.84
GeDi (Krause et al., 2020)	0.217	60.03	0.79	**0.62**	0.84	0.83
DExperts (Liu et al., 2021)	0.128	32.41	0.76	0.58	0.84	0.84
CriticControl	**0.081**	17.02	0.81	0.56	0.84	**0.87**

Table 4: Experimental results on the detoxification task. The results are compared with baselines for steering a freezed GPT-2-large model. We adopt the baseline results from (Liu et al., 2021).

Figure 6: Human preference test results for detoxification. This experiment shows comparisons of preferences between two continuations after the same prompts, CriticControl vs. GPT-2 and CriticControl vs. DExperts.

strate better performance than PPLM. Among them, GeDi outperforms CC-LM's control performance by generating guided texts through the use of contradicting positive and negative CC-LMs. However, the different text generation strategies of GeDi and CC-LMs lead to a reduction in text quality, as indicated by the perplexity score. On the other hand, CriticControl trains the critic in a sequential decision view and allows the actor and critic to share the same experience, resulting in the best performance on metrics. And both the topic and sentiment control experiments show the effectiveness of CriticControl in improving grammatical correctness compared to naive GPT-2. We explain that this is because CriticControl increases the amount of information within the region identified by the reward model, whereas naive GPT-2 does not. Furthermore, our additional experiment on PPO in Table 2 shows that CriticControl even improves the performance of RL-finetuned language models, not just freezing language models. Overall, our results show that CriticControl is promising for extending the use of RL in downstream tasks.

4.2.2 Human Preference Tests

For human evaluation, we conduct preference tests by comparing CriticControl with GPT-2-medium and GeDi. We randomly select 200 samples from the test set and ask annotators to indicate 1) Positiveness: Which sentences are more positive, and 2) Fluency: Which are more linguistically fluent. As in the topic control experiment, we take the majority vote of 3 annotators for each comparison.

Results As shown in Figure 5, CriticControl is successful at generating positive text compared to other baseline systems. However, since CriticControl should force negative prompts to be positive, the fluency of the generated text is poorer than that of a naive GPT. On the other hand, when we compare CriticControl to GeDi, we observe that CriticControl has much better fluency while still maintaining a high level of positivity. We find that people show the same preference for positivity among the two models and choose equality for considerable samples, different from the automatic evaluation results. We explain that this is because people have difficulty choosing more positive sentences among two positive sentences, unlike distinguishing between positive and negative sentences.

4.3 Detoxification

LMs might generate offensive or biased responses that are risky. To remedy this issue, we conduct experiments on reducing the toxicity of LMs as another controlled text generation task. We use *GPT-2-Large* as base LM and train the reward model using *BERT-based classification* (Devlin et al., 2018) models on a dataset from the Jigsaw Unintended Bias in Toxicity Classification Kaggle challenge [4]. We use the prompts in RealToxiciyPrompts dataset (Gehman et al., 2020) which consists of $100K$ prompts. During the training, we use $90K$ prompts in this dataset to train the critic. We use the remaining $10K$ non-toxic test prompts as DEx-

[4] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

perts and generate 20 tokens. We compare our approach with naive *GPT-2-Large*, PPLM, DAPT (Gururangan et al., 2020), GeDi, and DExperts (Liu et al., 2021). Detailed implementations are in Appendix A.3.

4.3.1 Metrics and Evaluation

We evaluate a total of $10k$ generated sentences for toxicity, fluency, and diversity. We use the Perspective API (Hosseini et al., 2017) [5] to classify sentences based on the most toxic token. We measure fluency and diversity as the same way in the previous section on topic control 4.1.

Results The experimental results in table 4 show that CriticControl outperforms other existing baselines. CriticControl effectively avoids using toxic language while maintaining the natural flow of the text. Since this task aims to remove toxic elements while preserving the amount of information, the diversity scores are all similar between baselines.

4.3.2 Human Preference Tests

For human evaluation, we conduct preference tests comparing CriticControl to GPT-2-large and DExperts. We randomly draw 200 samples from the results of the test set and ask annotators to indicate 1) Less Toxic: Which one is more rude or disrespectful (toxic comparison) and 2) Fluency: Which one is more grammatically correct and coherent? As in the topic and sentiment control experiment, we adopt the majority vote of three annotations for each comparison.

Results Human evaluation results in Figure 6 state that CriticControl is almost equivalent to the fluency of GPT-2 and exceeds the DExperts. Furthermore, the results verify that when evaluating the toxicity of the text generated from the same prompts, CriticControl consistently outperforms both GPT-2 and Dexperts.

4.4 Qualitative Analysis and Discussion

We analyze how the critic in our proposed method controls language models to generate the desired texts. As a qualitative analysis, we identify the probability change statistics of each word during the decoding step of CriticControl. We then discuss the most critic-favored and most critic-rejected words in each task. In topic and sentiment control, each word whose probability is quite improved through

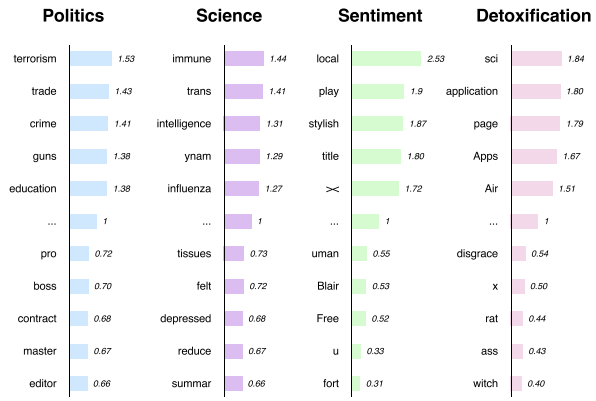

Figure 7: List of words promoted or rejected by Critic in relation to each task. The figure plots the probability ratio before and after control.

the critic is highly related to the given topic or sentiment. For example, '*terrorism, crime*, and *guns*' come up for political topics, '*immunity* and *flu*' for science topics, and '*play, stylish*, and *emoji* ><' for positive sentiment. One of the interesting findings is that the critic strongly rejects emotional words, such as '*felt* and *depressed*', to generate sentences suitable for science topics. We explain this because language models struggle to generate scientific texts starting from emotional prompts. We also observe another notable point to discuss in the detoxification experiment. The detoxifier-critic recommends words related to science or technology to the language model while rejecting offensive and insulting words. The frequently recommended words such as '*sci*, *application*, and *apps*' are factual words because the opportunity for the language model to make an aggressive remark disappears when a prompt for objective facts are created in the first place. These analyses verify how well CriticControl corrects the word's probability to achieve its goal in terms of language models.

5 Conclusion

We propose CriticControl, a controlled text generation method that takes advantage of both reinforcement learning and weighted decoding. Through experiments on various controlled text generation tasks, we demonstrate that CriticControl can effectively guide language models toward desired attributes while producing high-quality texts. Additionally, we show that CriticControl has a strong generalization ability in zero-shot attribute control tasks by using a general reward model. One of the limitations of this approach might be its high computational cost to explore with 'GPT3-scale' lan-

[5] https://perspectiveapi.com/

guage models (Brown et al., 2020), and we expect that this can be addressed through offline reinforcement learning (Fujimoto et al., 2019) techniques in future research.

Limitations

Large language models over the GPT-3 have made significant progress in natural language generation, but applying the CriticControl method, and exploring through these large language models are computationally too expensive. To address this, offline reinforcement learning (Fujimoto et al., 2019) may be a promising option to minimize training costs. CriticControl also has inference speed degradation because additional inference costs are needed like other controlled text generation methods (Dathathri et al., 2019; Yang and Klein, 2021). The potential solution may be to use the action-value predicting critic (Yue et al., 2020), which would allow for *real-time* control of various attributes without affecting the inference speed of the language model. Recently, the impact of instruction models (Chung et al., 2022; Ouyang et al., 2022) on text generation has recently been highlighted in academic research. These models, which allow for control over the generated text via input manipulation, have become widely accessible on various attributes without extra computational costs. Future works will investigate the synergistic potential between the *'input-side'* control of instruction-based models and the *'output-side'* control of CriticControl.

Ethical Statement

We acknowledge that our reward-driven text generation system may lead to generating harmful or misleading content when used with undesired reward models. However, controlled text generation methods have the potential to address these ethical issues present in large-scale pretrained language models, for example, through the detoxification of language. Therefore, we emphasize the proper use of reward models to pursue the public good and believe that it is important to continue research in this area as these techniques can offer significant benefits.

Acknowledgements

K. Jung is with ASRI, Seoul National University, Korea. This work has been financially supported by SNU-NAVER Hyperscale AI Center. This work was partly supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)]

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

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

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

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

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

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

Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR.

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

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

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. *arXiv preprint arXiv:1903.10671*.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. 2004. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(9).

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin. 2022. A distributional lens for multi-aspect controllable text generation. *arXiv preprint arXiv:2210.02889*.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric P Xing, and Zhiting Hu. 2021. Text generation with efficient (soft) q-learning. *arXiv preprint arXiv:2106.07704*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

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

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. *arXiv preprint arXiv:1805.06087*.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR.

Youngsoo Jang, Jongmin Lee, and Kee-Eung Kim. 2021. Gpt-critic: Offline reinforcement learning for end-to-end task-oriented dialogue systems. In *International Conference on Learning Representations*.

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

Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Dong Hyeon Jeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, et al. 2021. What changes can large-scale language models bring? intensive study on hyperclova: Billions-scale korean generative pretrained transformers. *arXiv preprint arXiv:2109.04650*.

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

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. *Advances in Neural Information Processing Systems*, 34:14542–14554.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Zhiyu Lin and Mark Riedl. 2021. Plug-and-blend: A framework for controllable story generation with blended control codes. *arXiv preprint arXiv:2104.04039*.

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. *arXiv preprint arXiv:2105.03023*.

Ximing Lu, Sean Welleck, Liwei Jiang, Jack Hessel, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *arXiv preprint arXiv:2205.13636*.

Tao Meng, Sidi Lu, Nanyun Peng, and Kai-Wei Chang. 2022. Controllable text generation with neurally-decomposed oracle. *arXiv preprint arXiv:2205.14219*.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. 2022. Mix and match: Learning-free controllable text generation using energy language models. *arXiv preprint arXiv:2203.13299*.

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

Khanh Nguyen, Hal Daumé III, and Jordan Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. *arXiv preprint arXiv:1707.07402*.

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

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

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. *arXiv preprint arXiv:2202.13257*.

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

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Ashish Sharma, Inna W Lin, Adam S Miner, David C Atkins, and Tim Althoff. 2021. Towards facilitating empathic conversations in online mental health support: A reinforcement learning approach. In *Proceedings of the Web Conference 2021*, pages 194–205.

Charlie Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine. 2022. Offline rl for natural language generation with implicit language q learning. *arXiv preprint arXiv:2206.11871*.

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

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. 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)*, pages 3269–3279, Hong Kong, China. Association for Computational Linguistics.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Bhargav Upadhyay, Akhilesh Sudhakar, and Arjun Maheswaran. 2022. Efficient reinforcement learning for unsupervised controlled text generation. *arXiv preprint arXiv:2204.07696*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

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.

Sanghyun Yi, Rahul Goel, Chandra Khatri, Alessandra Cervone, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2019. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. *arXiv preprint arXiv:1904.13015*.

Yuguang Yue, Yunhao Tang, Mingzhang Yin, and Mingyuan Zhou. 2020. Discrete action on-policy learning with action-value critic. In *International Conference on Artificial Intelligence and Statistics*, pages 1977–1987. PMLR.

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

A Implementation Details

All training on GPT-2-{small, medium, large, lx} is performed on two NVIDIA RTX A5000 24GB. The following implementation is based on temperature $T = 2$ in equation 1 and λ = 0.95 and γ = 0.99 in equation 2.

A.1 Topic Control

To control a language model for universal topics, we adopted *BART-Large-MNLI* (Lewis et al., 2019), language models for measuring universal relevance, as a general reward model rather than a binary classifier. When GPT-2 generates 80 continuations from the prompt, the reward model evaluates how relevant the generation is to the desired topic, and the critic learns to predict this relevance-reward. We follow the training setup of PPLM (Dathathri et al., 2019), training value predictor with seven topics (computers, space, military, legal, politics, science, and religion)). We take temperature $T = 2$ during sequential samplings, exploring various text trajectories. For inference, CriticControl gets repetition-penalty with a greedy decoding strategy. We follow previously reported human and automatic evaluation results of (Yang and Klein, 2021), including pure *GPT-2-medium* for naive generated texts, PPLM (Dathathri et al., 2019), WDEC and FUDGE (Yang and Klein, 2021). For the ablation study, we vary the size of the language model scales from *GPT-2-small* to *GPT-2-XL*.

A.2 Sentiment Control

Since this task is to manipulate the emotions of movie reviews, all baselines compare the controlled text from *GPT-2-medium* finetuned on IMDB movie review dataset. During training procedure, LM completes texts from GPT-2 without any shifts, starting with positive and negative IMDB prompts. Then, the critic observes these experiences and learns rewards generated from the reward model, *distilBERT* (Sanh et al., 2019) sentiment classifier finetuned on 2.5k reviews in IMDB dataset. Finally, CriticControl generates critic-guided text using nucleus sampling with a probability of 0.9 on test datasets. All baselines generate and compare their own 'guided' texts from the 2.5k test IMDB prompts. Our discussion starts from GPT-2 - medium, the basic baseline for sentiment control. For discussing PPLM, we retrain IMDB sentiment classifier for gradient updates. Then, PPLM decodes greedily on updated latent representation $H_t + \Delta H_t$. GeDi consists of *GPT-2-XL* and two polar generative discriminators, CC-LMs (Krause et al., 2020) finetuned on 'positive' or 'negative' IMDB movie reviews. For a fair comparison, we downgrade GeDi's language model to *GPT-2-medium* finetuned on IMDB. We discuss both 'Positive' CC-LM and GeDi for the sentiment control experiment. We add an experiment to answer the question, "Could reinforcement-learned models be critic-guided to achieve goals more appropriately?" To verify this potential, we finetune PPO (Schulman et al., 2017) on unfreezed *GPT-2-medium* by our reward model, and compare naive PPO and PPO with CriticControl.

A.3 Detoxification

In order to fairly compare our approach with other methods for controlled text generation, we use *GPT-2-Large* as our base language model. We train the reward model using *BERT-based classification* (Devlin et al., 2018) models on a dataset from the Jigsaw Unintended Bias in Toxicity Classification Kaggle challenge [6]. All evaluated generations start from the RealToxiciyPrompts dataset (Gehman et al., 2020), which consists of $100K$ prompts designed to elicit toxic responses. And we follow the experimental setup used by DExperts. During critic training, we use $90K$ toxic and non-toxic prompts from the train set with our reward model. For evaluation, we use the same $10K$ non-toxic test prompts as DExperts and generate 20 tokens using top-10 sampling. We also include reported baselines from DExperts (Liu et al., 2021), including naive *GPT-2-Large*, PPLM, DAPT (Gururangan et al., 2020), GeDi, and DExperts.

B Human Evaluation

During human evaluation, the end of the sentence is almost incomplete because baselines generate a fixed number of tokens from the prompt. Therefore, we added '[...]' for the end of all generated texts for comfortable reading. Then, we give instructions about 'Don't care about the incomplete last sentence for evaluating fluency, which is marked with [...] due to length limitation.' to annotators as in Figure 8. Also, we notice 'Please select equal only when it is really difficult to judge' for more accurate human evaluation results in the preference test. We pay MTurk workers a competitive pay of more than $10

[6] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

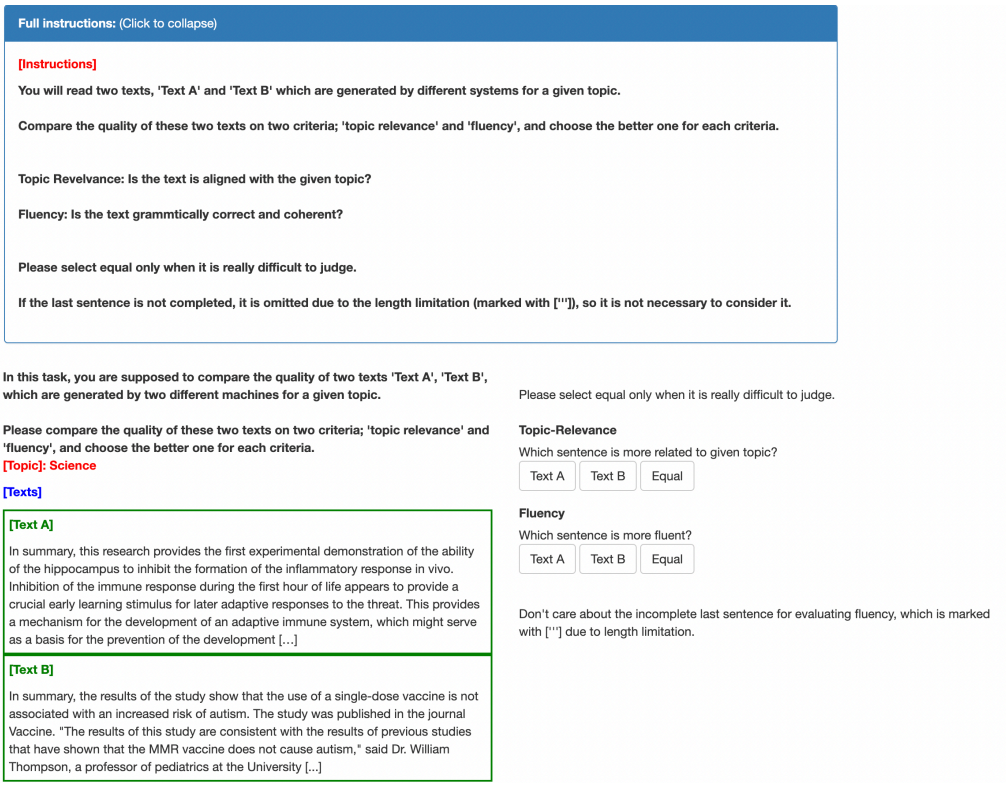

Figure 8: An example instruction page shown to Amazon MTurk annotators for human preference test on the topic control.

an hour. We hire the workers whose nations in one of the US, CA, UK, AU, NZ. We restrict the annotators whose HIT minimum hits are over 1000 and HIT rates are higher than 96%.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 'Limitations'

☑ A2. Did you discuss any potential risks of your work?
Section 'Ethical Considerations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 'Abstract' and '1. Introduction'

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section '4 Experiments'

☑ B1. Did you cite the creators of artifacts you used?
Section '4 Experiments'

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No, all artifacts we used are based on open-source and have no restriction for research

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section '4 Experiments' and 'Appendix'

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section '4 Experiments' and 'Appendix'

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section '4 Experiments' and 'Appendix'

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section '4 Experiments' and 'Appendix'

C ☑ Did you run computational experiments?

Section '4 Experiments'

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 'Appendix'

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 'Appendix'

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section '4.4 Qualitative Analysis and Discussion'

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section '4 Experiments'. We report used pre-trained models served by Huggingface

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section '4 Experiments' and 'Appendix'

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section 'Appendix'

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section '4 Experiments' and 'Appendix'.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Section '4 Experiments' and 'Appendix'.

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Section '4 Experiments

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Section'Appendix'