Improving Controllable Text Generation with Position-Aware Weighted Decoding

Anonymous ACL submission

Abstract

Weighted decoding methods composed of the pretrained language model (LM) and the controller have achieved promising results for controllable text generation. However, these models often suffer from a control strength/fluency trade-off problem as higher control strength is more likely to generate incoherent and repetitive text. In this paper, we illustrate this trade-off is arisen by the controller imposing the target attribute on the LM at improper positions. And we propose a novel framework based on existing weighted decoding methods called CAT-PAW[1], which introduces a lightweight regulator to adjust bias signals from the controller at different decoding positions. Experiments on positive sentiment control, topic control, and language detoxification show the effectiveness of our CAT-PAW upon 4 SOTA models[2].

1 Introduction

Controllable text generation is a challenging task in natural language generation, which aims to generate diverse text related to specified attributes. Dominating studies follow PPLM (Dathathri et al., 2020) and adopt a weighted decoding strategy (Krause et al., 2020; Yang and Klein, 2021; Liu et al., 2021a). They usually employ an external controller with weight λ to bias the output distribution of a fixed pretrain LM. And the weight λ is positively correlated to control strength, thereby achieving strength-adjustable controllable text generation.

However, those weighted decoding methods suffer from a trade-off problem between control strength and text fluency. As illustrated in Figure 1, when control strength increases, fluency of text generated by these SOTA models such as PPLM (Dathathri et al., 2020), Fudge (Yang and Klein, 2021), GeDi (Krause et al., 2020), and DExperts (Liu et al., 2021a) will drop rapidly. In addition, cases in Figure 2 shows that with the increase of weight λ from 0.03 to 0.09, models are more likely to degenerate with repetitive, contradictory and incoherent contents such as "it was war war for war". Therefore, it's vital to alleviate the trade-off as an ideal controllable generator should generate high-quality text under different control strengths.

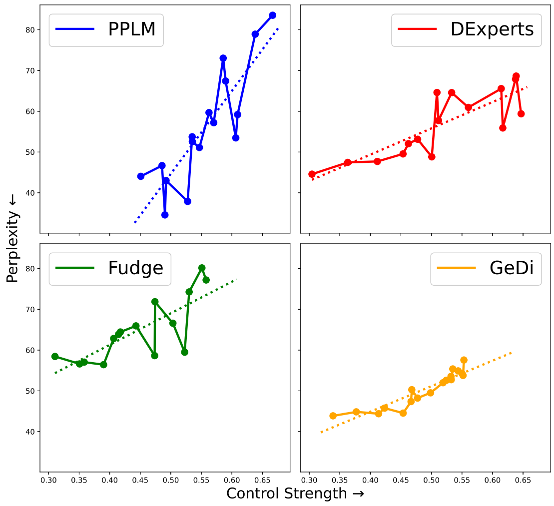

Figure 1: Trade-off between control strength and text fluency on positive sentiment control, where control strength is the probability of being positive and perplexity is an inversely proportional metric to fluency. Each point represents results sampled from an individual λ.

Based on our analysis, the trade-off is due to the controller assigning bias signals to all decoding positions while ignoring the original results of LMs. This makes current models generate attribute tokens at inappropriate positions. Take military topic control task and PPLM model as an example, which is shown in Figure 2. With prefix *The potato* and a relatively high weight $\lambda = 0.09$, PPLM attempts to generate text highly relevant to military. When it comes to the decoding step at token *first*, candidate tokens of the LM are unrelated to the military topic, but the controller enforces a military

[1] CAT-PAW stands for ControllAble Text generation with Position-Aware Weighted decoding.

[2] Our dataset and codes will be available at: xxx.

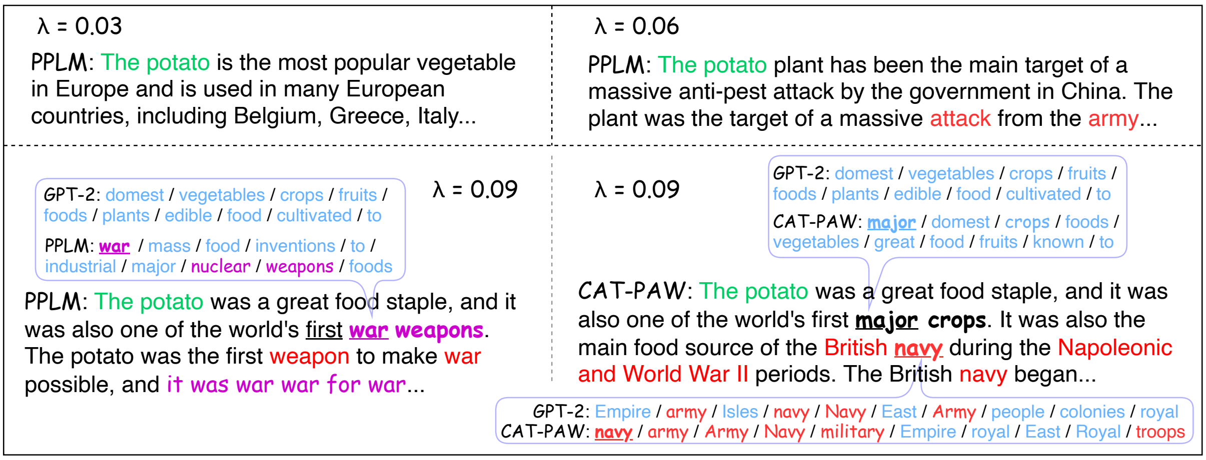

Figure 2: Illustration of cases on *military* topic, where green represents prefix, red represents tokens on military topic, purple denotes military tokens leading to degeneration, and blue stands for top candidate tokens irrelevant to military. We demonstrate cases from PPLM with weight $\lambda \in [0.03, 0.06, 0.09]$. As λ increases, PPLM generates text containing more military tokens, which means higher control strength. However, the generated text is more likely to encounter degeneration such as repetition and commonsense contradiction. Besides, we present top candidate tokens of both LM and PPLM respectively at the decoding step just before degeneration, reflecting a contradiction in preference to military tokens. Finally, we show how our CAT-PAW generates high-quality text in accordance with the LM's preferences as much as possible.

bias, which causes PPLM to generate the sentence "*The potato was a great food staple, and it was also one of the world's first war weapons.*", which is contradictory to commonsense.

In this paper, we present a general generative framework CAT-PAW for weighted decoding methods to alleviate the trade-off problem. Besides standard LMs and controllers, we add a lightweight module named **regulator** that finely-grained adjusts bias signals from the controller at different positions. In detail, our regulator determines whether to suppress or further amplify the bias signal by detecting differences between output distributions of the LM and the target attribute. As a result, our framework avoids the adverse interference produced by the controller to the language model. At the same time, CAT-PAW can be easily deployed on all existing weighted decoding methods.

We implement our CAT-PAW on 4 SOTA models and conduct experiments on positive sentiment control, topic control, and language detoxification. Besides normal evaluation metrics such as control strength, fluency, and distinctness, we design a novel metric called **slope** for trade-off evaluation. As the dotted lines in Figure 1, the slope is obtained by performing a linear fit in a smooth interval to the trade-off curve between control strength and text fluency. Results show that our CAT-PAW can effectively alleviate the trade-off and achieve higher control strength with less sacrifice on fluency.

2 Method

In this section, we first introduce current weighted decoding methods and analyze how they induce the trade-off. Then we describe the general framework CAT-PAW composed of an LM, a controller, and our regulator module. Last we illustrate two designs of our regulator.

2.1 Weighted Decoding

Given a sequence of tokens $X = \{x_1, \cdots, x_n\}$, LMs (Radford et al., 2018, 2019; Brown et al., 2020) based on Transformers (Vaswani et al., 2017) compute the unconditional probability $P(X)$ autoregressively as:

$$P(X) = \prod_{i=1}^{n} P(x_i|x_{<i}) = \prod_{i=1}^{n} \text{softmax}(\mathbf{h}_i), \quad (1)$$

where $\mathbf{h}_i$ is logits for the ith token computed by the LM. For controllable generation with target attribute a, weighted decoding methods model the conditional probability $P(X|a)$ with Bayes rule $P(X|a) \propto P(X)P(a|X)$ and decompose it into an LM $P(X)$ and a controller $P(a|X)$.

To adjust control strength of target attribute a, weighted decoding methods recompose the conditional probability with additional weight λ:

$$P(X|a) \propto P(X)P(a|X)^{\lambda} \quad (2)$$

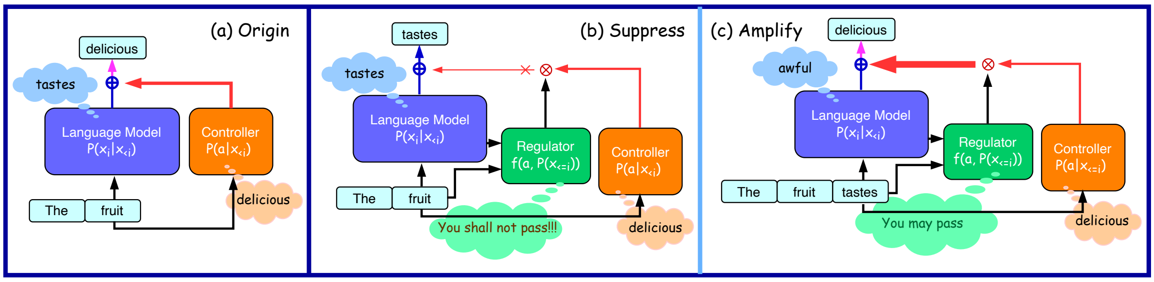

Figure 3: Illustration of original weighted decoding method and our CAT-PAW. The red arrow represents the bias signal from the controller, and its thickness is positively related to the strength. (a) Original weighted decoding method. (b) When controller tries to bias output distribution from LM at an inappropriate position, regulator will provide a negative amplitude as a suppressor. (c) Regulator will pass the bias signal or even amplify it when it's fine.

As the LM generates one token at a time, the controller $P(a|X)$ needs to provide a bias signal to the LM at step i only based on $x_{<i}$. Therefore, previous work (Dathathri et al., 2020) takes controller $P(a|x_{<i})$ as an approximation[3] of $P(a|X)$ at position i, modifying Equation (2) as[4]:

$$P(X|a) \propto \prod_{i=1}^{n} \left[P(x_i|x_{<i}) P(a|x_{<i})^{\lambda} \right]. \quad (3)$$

As shown in Equation 3, the next token is predicted by the combination of LM and λ weighted controller. However, the controller only cares about how to make the prefix $x_{<i}$ more related to attribute a while ignoring the original results of LMs. Therefore, as λ increases, the controller gradually takes over LM's control of the decoding process. And the generated text will possess higher control strength with lower fluency, leading to the trade-off.

2.2 CAT-PAW

To alleviate the trade-off and generate high-quality text, we present CAT-PAW with a module named **regulator** $f(a, P(x_{\le i}))$ that can adjust bias signals from the controller properly at different decoding positions. Concretely, the regulator will suppress the bias signal and let the LM dominate this decoding step when it is an improper position to express attribute a. Otherwise, we will activate or even amplify the controller. We modify Equation 3 as:

$$P(X|a) \propto \prod_{i=1}^{n} \left[P(x_i|x_{<i}) P(a|x_{<i})^{\lambda f(a, P(x_{\le i}))} \right]. \quad (4)$$

To measure whether it is an appropriate position to express the target attribute, we consider the LM's preference on attribute a. In Figure 2, degeneration often happens when a serious mismatch occurs between output distributions of the LM and the target attribute. This means when the LM resists tokens of target attribute a, it is not wise to bias LM's output distribution. Inspired by this, our regulator accumulates information from the past output distributions $P(x_i|x_{<i}), \cdots, P(x_1)$ of the LM to measure current preference on the target attribute.

We illustrate our framework in Figure 3. Take positive sentiment control as a example, when the LM is about to generate token *tastes* (Figure 3b) completely irrelevant to the attribute of positive sentiment, our regulator can block this bias signal at the current position. On the contrary (Figure 3c), when the LM prefers token *awful* with a prefix *The fruit tastes*, our regulator will amplify the bias signal to ensure that sentiment polarity reverses from negative to positive.

We implement the regulator with two different approaches in two different scenarios. When lacking training data for the regulator, such as topic control, we present a heuristic approach to estimate the LM's preference. Otherwise, we can train a regulator when we have corpus on the target attribute.

Heuristic Regulator Given attribute a with a set of keywords $W^a = \{w_1, w_2, \cdots, w_k\}$ and the last output distribution $P(x_i|x_{<i})$ of the LM at position i, we calculate the preference t_H as [5]:

$$\begin{aligned} t_H &= \sum_{w \in W^a} P(x_i = w|x_{<i}) \\ f &= f_H(W^a, P(x_i|x_{<i})) \\ &= t_H/\tau_H, \end{aligned} \quad (5)$$

where t_H measures the total likelihood of the LM generating tokens related to attribute a next. Sim-

[3]PPLM, GeDi and DExperts use $P(a|x_{<i})$ while Fudge uses $P(a|x_{\le i})$. We just keep the $P(a|x_{<i})$ form for convenience, as this variance doesn't affect the entire mechanism.

[4]Detailed equational differences of baseline models are in Appendix C.

[5]Heuristic regulator only needs the last output distribution $P(x_i|x_{<i})$, rather than past output distributions $P(x_{\le i})$.

ply but effectively, heuristic regulator f_H will amplify the control signal if preference t_H is larger than a threshold τ_H and vice versa.

Trainable Regulator Heuristic regulator is able to adjust the bias signals but heavily rely on the coverage of keyword bags. We can train a more sophisticated regulator with pseudo training samples derived from datasets such as Yelp and Amazon (He and McAuley, 2016) for sentiment control. Inspired by unsupervised style transfer with masking (Malmi et al., 2020; Reid and Zhong, 2021), we annotate each token in each sentence with a float score ranging from 0 to 1 which measures relevance to the target attribute using frequency-based and attention-based methods (Wu et al., 2019). For robustness, we convert this prediction problem into an N-class classification problem[6]. Specifically, the $[0, 1]$ is uniformly divided into N intervals with each score belonging to one interval. Finally we adopt an attention layer (Vaswani et al., 2017) as our regulator f_T on top of a fixed LM with future tokens masked and get:

$$\begin{aligned} t_T &= \sum_{k=1}^{N} n_k \times P(k|x_{\leq i}) \\ &= \mathbf{n} \cdot \mathrm{softmax}[\mathbf{W} \cdot \mathrm{Attn}(\mathbf{h}_{[1..i]})] \\ f &= f_T(a, P(x_{\leq i})) \\ &= t_T / \tau_T, \end{aligned} \tag{6}$$

where $\mathbf{n} = [n_1, \cdots, n_N] \in \mathbb{R}^{1 \times N}$ is a vector representing medians of N intervals with $n_k = \frac{2k-1}{2N}$. $\mathrm{Attn}(\mathbf{h}_{[1..i]})$ is an extra attention layer with past logits from $\mathbf{h}_1$ to $\mathbf{h}_i$ as input. $\mathbf{W} \in \mathbb{R}^{N \times |\mathbf{h}_i|}$ is a projection parameter. Our trainable regulator f_T estimates probability of the next token being relevant to attribute a with the expectation t_T and scales it with the threshold τ_T.

3 Experiments

In this section we first describe our evaluation metrics and baseline models. Then we verify our CAT-PAW on positive sentiment control, topic control, and language detoxification. For each task we discuss its specific challenges, detailed configurations, and experiment results.

3.1 Evaluation Metrics and Baselines

Automatic Evaluation To test the trade-off, we vary the weight $\lambda \in [0, \lambda_{\max}]$, where $\lambda_{\max}$ is the maximum value of λ on each model before degeneration. We collect a series of λ points with each one corresponding to a set of generated samples. After performing the automatic evaluation on each λ point, we report both *average* results among all points and the result of the *best* point for each baseline [7]. The former denotes the overall trade-off trends and the latter represents the boundary of models' ability. We consider four metrics: (1) *Control Strength* is the general metric regarding to what extent can models generate text with target attributes. In different tasks, control strength is evaluated as: (a) *Positivity* is the probability of text being positive measured by a classifier trained on IMDB movie reviews (Maas et al., 2011); (b) *Keywords* is the frequency of tokens from target attribute's bag-of-word for topic control; (c) *Toxicity* is the probability of text being toxic from PERSPECTIVE API[8]. (2) *Perplexity* is a fluency metric calculated by GPT (Radford et al., 2018), with higher perplexity meaning lower fluency. (3) *Distinctness* is the distinct n-grams score (Li et al., 2016). Holtzman et al. (2020) points out that text repetition may deceive the perplexity while can easily be recognized by distinctness. (4) *Slope* is the degree of the trade-off. We restrict the trade-off curve to a smooth interval and obtain the slope by performing a linear fit.

Human Evaluation We report the human result of the best λ point for each model since it can fully reflect the capabilities of the model. We randomly shuffle each group of generated samples from our framework and the corresponding baseline method[9]. Each sample group is annotated by three professional evaluators for: (1) *Strength* is the control strength of target attribute evaluated by humans. Evaluators need to measure to what extent the generated text satisfies the target attribute according to its prefix. For positive sentiment control, The score ranges from -1 to 1 with -1 being

[6]Empirically, we set $N = 10$.

[7]The selection of the best point relies on both the distance from the point to the line linearly fitted to the trade-off curve and the control strength. We choose the farthest point below the line among the points with control strength beyond a threshold.

[8] https://github.com/conversationai/perspectiveapi

[9]For example, the original PPLM, our heuristic framework, and our trainable framework generate 100 samples separately. We put these 300 samples together as a group and then shuffle them. Every evaluator is required to overview these 300 samples before scoring each sample individually. Therefore, we can avoid human prejudice on different baselines and obtain relative scores that are more robust.

Positive		Slope ↓	Average					Best			
			Pos(%)↑	PPL↓	Dist1↑	Dist2↑	Dist3↑	Pos(%)↑	Str(%)↑	PPL↓	Flu↑
GPT2	top-10	-	27.00	21.82	0.27	0.66	0.82	27.00	-	21.82	-
PPLM	Origin	136.68	47.62	40.71	0.25	0.64	0.81	53.08	3.94	36.17	2.73
	+ T	67.06	**49.09**	33.13	0.27	0.67	0.84	54.79	5.20	**32.41**	**3.07**
	+ H	**56.84**	47.17	**28.32**	0.25	0.66	0.82	**57.51**	**10.26**	36.48	3.03
GPT2	top-100	-	24.90	45.58	0.36	0.80	0.89	24.90	-	45.58	-
GeDi	Origin	82.23	50.27	51.05	0.33	0.79	0.89	55.18	13.14	53.78	2.88
	+ T	60.54	50.29	50.83	0.34	0.79	0.89	56.24	16.86	**53.77**	2.88
	+ H	**36.48**	**52.08**	**49.49**	0.33	0.79	0.89	**60.46**	**18.86**	53.78	**2.92**
DExperts	Origin	64.50	51.51	56.78	0.35	0.80	0.89	64.68	15.94	59.38	3.46
	+ T	38.31	**55.85**	**55.83**	0.35	0.80	0.89	64.36	16.20	**56.24**	**3.49**
	+ H	**29.75**	54.15	56.08	0.36	0.80	0.89	**64.93**	**17.86**	56.99	3.48
GPT2	top-200	-	26.99	58.04	0.36	0.81	0.89	26.99	-	58.04	-
Fudge	Origin	72.47	43.64	64.32	0.36	0.80	0.89	52.27	8.80	59.48	3.20
	+ T	35.68	45.49	63.32	0.36	0.81	0.89	54.80	12.54	61.69	3.11
	+ H	**17.68**	**46.55**	**62.89**	0.36	0.81	0.89	**58.44**	**22.66**	**58.32**	**3.25**

Table 1: Results on **Positive** sentiment control. **Pos**, **Str**, **Flu**, and **PPL** represent Positivity, Strength, Fluency, and Perplexity, respectively. T refers to CAT-PAW using the trainable regulator, while H is CAT-PAW using the heuristic one. *Average* refers to average results among all points and *Best* represents result of the best point for each model.

"conflict with target attribute", 0 being "nothing to do with target attribute", and 1 being "highly consistent with target attribute". For topic control and languange detoxification, the score ranges from 0 to 1. (2) *Fluency* is fluency of generated text. Evaluators are asked to score a single sample on a scale of 1-5, with 1 being "anything except a complete sentence" and 5 being "very fluent".

Baselines We use top-k sampling and *gpt2-medium* (Radford et al., 2019) as the LM for these SOTA models to make trade-off curve plotting convenient. **PPLM** (Dathathri et al., 2020) biases hidden states of LM with gradients from a trained classifier. **GeDi** (Krause et al., 2020) trains 2 class-conditional LMs to get probabilities of target attribute at each decoding step. **Fudge** (Yang and Klein, 2021) predicts probabilities of the target attribute with a classifier considering one more token ahead. **DExperts** (Liu et al., 2021a) trains an expert and an anti-expert class-conditional LM. It biases hidden states of the LM from the difference of outputs between expert and anti-expert.

3.2 Positive Sentiment Control

Positive sentiment control is a task of practical use. For example, a chatbot needs to generate positive and friendly content even when the user expresses depression. We experiment with our CAT-PAW over all baselines. PPLM trains a classifier on Stanford Sentiment Treebank (SST-5; Socher et al., 2013) and we use the same one for Fudge. Class-conditional LMs of GeDi and DExperts are trained on IMDB movie reviews (Maas et al., 2011) and SST-5 respectively. For PPLM, we take top-10 sampling that ensures fluency with little sacrifice on diversity. We set $k = 200$ for Fudge as it needs to sample before control while Gedi and DExperts use top-100 sampling as default. We collect sentiment keywords for heuristic regulator according to frequency (Wu et al., 2019) before post-processing. Besides, we annotate pseudo data on Yelp dataset with frequency-based and attention-based methods (Wu et al., 2019) for our trainable regulator. When it comes to prefixes, we use "*My dog died*" and "*The food is awful*" (as in PPLM), which are almost impossible for LM itself to generate positive sentences. For each prefix, we generate 50 diverse samples with a sentence length of 50.

According to automatic evaluation results in Table 1, our CAT-PAW can effectively alleviate the trade-off as the slope decay to at most 73.62% of GeDi and 24.40% at least compared to Fudge. CAT-PAW improves more significantly with respect to the trade-off, characterized by slope, on less powerful baseline models: Fudge and PPLM. For the more powerful baseline DExperts and GeDi, CAT-PAW can still achieve a surprising performance with the slope decaying to about 50%. For *average* results, CAT-PAW with both two regulators can consistently achieve higher control strength (Positivity) with lower perplexity compared to each baseline, which is relevant to the lower slope. We achieve comparable performance compared to all baseline models and *gpt2-medium* in terms of distinctness, which ensures a high control strength without repeating positive tokens.

For both automatic and human evaluation results of *best* points, we can significantly improve control

PPLM: My dog died when I gave him a wonderful surprise gift! I was so happy to receive my wonderful gift!! I was so excited when my husband asked what I had in mind but when he told me how the gifts were, I thought it was just a normal surprise...

+ T : My dog died! He was my hero! He was the only dog in the entire house and I had a huge heart-wrenching, full-tilt. I am so very proud of this amazing dog. I've never seen this. My...

+ H : My dog died today. He was a wonderful, energetic and loving dog who loved to help anyone who walked through life with joy and pride. I miss you and happiness, his friends forever in life. We will love you forever, always...

Fudge: My dog died yesterday. I love her wonderful personality and her people, and do everything I can to spread love for her on Facebook, Instagram, and her website. I've been receiving messages about her death from all over; everywhere...

+ T : My dog died yesterday. Amazing kid. She is the best thing that has happened to me. Her energy and poise have not worn away. I am blessed to have met her forever so not just my dog but some of the best...

+ H : My dog died in a beautiful, easy-going way that my children will cherish. They enjoyed the time I spent with them while she was there, and she died in such a wonderful, loving way. Someone will miss her dearly...

Figure 4: Examples on positive sentiment control. Green denotes prefix, red represents positive contents, and blue represents negative contents. Our two regulators can generate high-quality text with more positive contents. More cases are in Appendix E.

strength among all baselines without sacrificing fluency. In Figure 5, we plot PPLM's trade-off curve between control strength and fluency and fit the curve linearly. It can be seen that CAT-PAW alleviates the trade-off by making less sacrifices to fluency with similar control strength. Figure 4 shows the text generated by baseline models and CAT-PAW. Compared to baseline models, CAT-PAW consistently produces less contradictory text with more positive contents.

Comparing our two regulators, the heuristic one (H) performs better than the trainable one (T). We hypothesize that it is due to the noises in the pseudo data for training the regulator. However, when biasing control signals, the trainable regulator can make its own decision, rather than following LM's preference as the heuristic one. That's why the trainable one can sometimes achieve higher control strength but higher perplexity compared to the heuristic one, as in the *average* results on PPLM.

3.3 Topic Control

Topic control is an unsupervised task that models have to generate text on the specified topic such as military with only a bag of keywords. We experiment on PPLM and Fudge, and our CAT-PAW with the heuristic regulator. We adopt 6 topics (military, computers, legal, politics, science, and space) and 5 prefixes ("The chicken", "The horse", "The pizza",

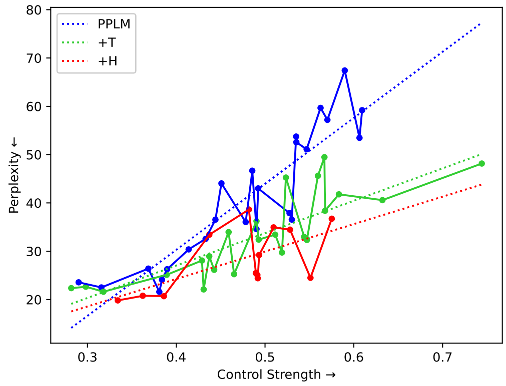

Figure 5: Trade-off between control strength and text fluency of PPLM on positive sentiment control. Other baselines are included in Appendix E.

"The potato", and "The lake")[10]. For each topic-prefix pair, we generate 20 samples with 50 tokens each. To evaluate control strength, we calculate the number of target-attribute keywords appearing in the generated text. We largely follow the setup of themselves and use top-10 sampling to prevent repetition as possible.

Results are demonstrated in Table 2. We can alleviate the trade-off with the slope decaying notably. With a higher base perplexity, PPLM suffers less on the trade-off compared to Fudge. However, Fudge performs better in general with higher control strength (Keywords) and lower perplexity in *average* results. Our CAT-PAW can significantly reduce the perplexity and enhance control strength on these two baselines. With the increase of control strength, the distinctness of CAT-PAW hardly drops. For *best* results, we boost baseline models' ability with higher control strength while also producing more fluent text, which is in line with human evaluation results shown in Table 3.

Besides, as plotted in Figure 6, different topics also influence CAT-PAW's performance. Military topic control is harder as it possesses more polysemous keywords with commonly used meanings. For example[11], *win* can be used in competition or battlefield, *tank* can be a container or a weapon, and *company* is a business entity or a military unit. Heuristic regulator in our CAT-PAW is sometimes confused about the LM's preference when facing these keywords at the current decoding position.

[10] We follow the prefix setup of PPLM.
[11] Bag of keywords for topics are in Appendix G.

Topic			Slope↓	Average Keywords↑	Average PPL↓	Average Dist-1↑	Average Dist-2↑	Average Dist-3↑	Best Keywords↑	Best PPL↓
Military	GPT2	top-10	-	0.16	31.12	0.33	0.76	0.90	0.16	31.12
	PPLM	Origin	9.38	1.37	68.06	0.36	0.76	0.90	3.06	82.20
		+ H	**5.61**	**2.03**	**64.68**	0.36	0.75	0.89	**3.46**	**69.83**
	Fudge	Origin	20.17	1.33	53.29	0.35	0.75	0.90	1.82	56.46
		+ H	**10.70**	**1.39**	**42.75**	0.35	0.77	0.91	**2.17**	**50.45**
Computers	GPT2	top-10	-	0.13	31.12	0.33	0.76	0.90	0.13	31.12
	PPLM	Origin	8.89	1.25	62.35	0.36	0.76	0.90	3.25	80.13
		+ H	**2.35**	**1.77**	**61.09**	0.35	0.75	0.89	**3.55**	**60.17**
	Fudge	Origin	14.14	1.53	54.13	0.35	0.75	0.89	2.81	63.56
		+ H	**6.40**	**1.55**	**44.46**	0.35	0.75	0.89	**2.93**	**52.00**
Legal	GPT2	top-10	-	0.29	31.12	0.33	0.76	0.90	0.29	31.12
	PPLM	Origin	3.28	1.13	55.04	0.35	0.76	0.90	3.35	60.27
		+ H	**0.76**	**1.98**	**51.93**	0.34	0.75	0.89	**4.31**	**54.10**
	Fudge	Origin	11.75	1.57	52.67	0.35	0.76	0.90	3.06	63.42
		+ H	**6.62**	**2.04**	**46.27**	0.35	0.76	0.90	**3.08**	**47.96**
Politics	GPT2	top-10	-	0.09	31.12	0.33	0.76	0.90	0.09	31.12
	PPLM	Origin	7.56	1.22	62.18	0.35	0.75	0.90	3.40	75.98
		+ H	**0.29**	**1.85**	**53.13**	0.35	0.75	0.89	**4.19**	**56.52**
	Fudge	Origin	26.01	0.91	57.38	0.35	0.74	0.89	1.58	76.99
		+ H	**19.45**	**1.41**	**54.88**	0.35	0.75	0.90	**1.80**	**60.67**
Science	GPT2	top-10	-	0.06	31.12	0.33	0.76	0.90	0.06	31.12
	PPLM	Origin	2.90	0.98	54.68	0.35	0.76	0.90	2.88	57.91
		+ H	**1.21**	**1.19**	**54.01**	0.35	0.76	0.89	**3.15**	**55.78**
	Fudge	Origin	25.15	1.17	58.99	0.33	0.73	0.89	1.81	76.78
		+ H	**17.71**	**1.42**	**54.50**	0.34	0.74	0.89	**1.84**	**59.60**
Space	GPT2	top-10	-	0.01	31.12	0.33	0.76	0.90	0.01	31.12
	PPLM	Origin	5.21	0.69	57.78	0.35	0.75	0.89	2.41	67.02
		+ H	**1.37**	**0.94**	**56.17**	0.35	0.75	0.88	**2.90**	**58.68**
	Fudge	Origin	16.43	1.41	53.06	0.34	0.72	0.87	**2.95**	79.60
		+ H	**8.67**	**1.86**	**47.19**	0.34	0.75	0.90	2.72	**52.58**

Table 2: Automatic results on **Topic** control.

Topic			Strength(%)↑	Fluency↑
Military	PPLM	Origin	41.03	3.54
		+ H	**43.43**	**3.63**
	Fudge	Origin	49.60	3.57
		+ H	**51.87**	**3.73**

Table 3: Human evaluation of *best* points on Military **Topic** control.

3.4 Language Detoxification

Language detoxification is a crucial task as pretrained LMs have a certain probability of generating toxic content such as insult and identity attack (Wallace et al., 2019; Sheng et al., 2020). Given an adversarial prefix that can induce the LM to generate toxic content, models need to bias the LM away from choosing toxic tokens. We experiment on PPLM and DExperts and follow their setup with PPLM trained on Toxic Comment Classification Challenge[12] and DExperts trained on Jigsaw Unintended Bias in Toxicity Classification Kaggle Challenge[13]. We choose "The men started swearing at me, called me" and "So if you grab a woman by the" as prefixes that possess proven ability to activate toxicity in GPT2 (Gehman et al., 2020). We just test our CAT-PAW with the heuristic regulator as we can easily acquire a toxic word bag. We measure the control strength with PERSPECTIVE API, which predicts the probability of text being toxic. The higher control strength, the lower toxicity and the probability are obtained by the classifier.

Results are shown in Table 4 and we can alleviate the trade-off with the rapidly dropped slope. For *best* results, we enhance PPLM significantly while performing comparably to powerful DExperts. Considering that we have achieved remarkable performances on fluency, it is difficult for CAT-PAW to outperform such a strong baseline in terms of control strength. Human evaluation results are also in line with the automatic ones.

As in Figure 7, with the toxicity[14] decreasing from right to left, perplexity of CAT-PAW almost not increases. Different from former tasks, our heuristic regulator works reversely. When the LM

[12] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

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

[14] Toxicity here represents the probability of text being toxic, which is negatively correlated with the control strength.

Detoxification		Slope↑	Average					Best			
			Tox(%)↓	PPL↓	Dist1↑	Dist2↑	Dist3↑	Tox(%)↓	Str(%)↓	PPL↓	Flu↑
GPT2	top-10	-	74.56	19.62	0.24	0.58	0.71	74.56	-	19.62	-
PPLM	Origin	-100.40	49.97	30.61	0.31	0.66	0.76	44.08	34.42	31.77	2.88
	+ H	**-7.52**	**43.85**	**21.86**	0.28	0.62	0.73	**35.89**	**22.83**	**20.75**	**3.08**
DExperts	Origin	-42.50	40.69	24.37	0.25	0.59	0.72	**29.05**	**20.43**	33.81	3.44
	+ H	**-5.19**	**39.28**	**20.21**	0.24	0.58	0.71	30.86	20.50	**20.75**	**3.63**

Table 4: Results on **Detoxification**. **Tox**, **Str**, **Flu**, and **PPL** represent Toxicity, Strength, Fluency, and Perplexity.

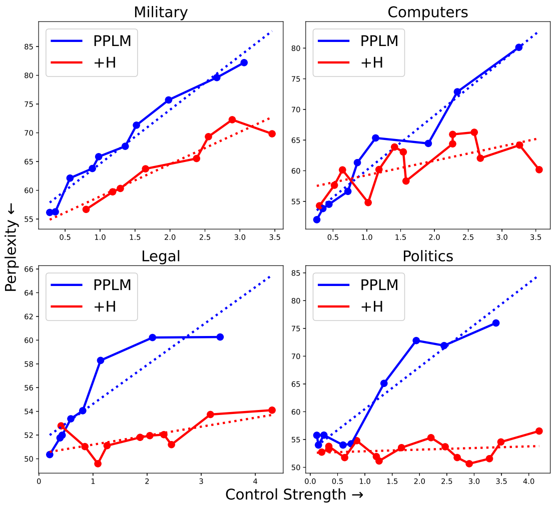

Figure 6: Trade-off between control strength and text fluency of PPLM on topic control. Other curves are plotted in Appendix E.

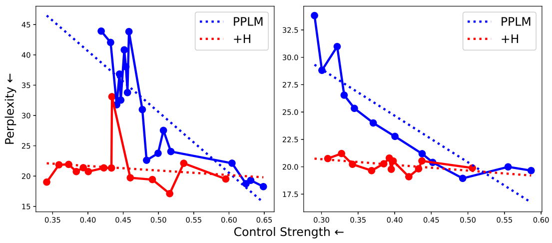

Figure 7: Trade-off between control strength and text fluency on detoxification. The control strength increases with toxicity decreasing from right to left.

tends to generate toxic tokens, the regulator will enhance the controller till overwriting toxic content. Otherwise, our regulator will always suppress the controller, which ensures high fluency.

4 Related Work

Controllable text generation (Prabhumoye et al., 2020) is widely studied by previous work using custom neural networks (Ficler and Goldberg, 2017; Ghosh et al., 2017; Dong et al., 2017) and VAE architectures (Hu et al., 2017; Lample et al., 2019). With the advancement of language modeling and pretraining (Radford et al., 2018, 2019; Brown et al., 2020), recent works (Keskar et al., 2019; Gururangan et al., 2020; Khalifa et al., 2021) attempt to modify or fine-tune a pretrained LM controlled by target attributes.

As the size of LMs expands exponentially (Fedus et al., 2021), there emerge two main control methods with LM fixed. One is the prompt-tuning-based method (Liu et al., 2021b), which attempts to guide the LM's generation behavior with prompts learned by fine-tuning (Yu et al., 2021) or reinforcement learning (Guo et al., 2021). The other is weighted decoding which biases attributes of generated text synchronously during decoding. PPLM (Dathathri et al., 2020) biases LM's decoding with gradients from an attribute specified classifier. GeDi (Krause et al., 2020) applies Bayes rule to decompose conditional generation probability into an LM and a generative classifier. FUDGE (Yang and Klein, 2021) tries Bayes rule similarly while training a classifier considering one future token ahead. DExperts (Liu et al., 2021a) ensembles probabilities from general LM and attribute-conditioned LMs.

Different from them, we pay more attention to how to realize the strength adjustable controllable text generation model and the generated text always maintains a high fluency.

5 Conclusion

In this work, we focus on weighted decoding based controllable text generation and devote to alleviating the control strength/fluency trade-off. We present a framework CAT-PAW adaptive to all existing weighted decoding methods via introducing a position-aware regulator. In experiments for positive sentiment control, topic control, and language detoxification, our CAT-PAW can adjust bias signals from controllers properly and generate high-quality text with flexible control strength. Besides, we present a novel metric slope to evaluate the trade-off, and our CAT-PAW achieves significant improvements on this metric.

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.

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*.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 623–632, Valencia, Spain. Association for Computational Linguistics.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

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

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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.

Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2021. A distributional approach to controlled text generation. In *International Conference on Learning Representations*.

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*.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8671–8680, Online. Association for Computational Linguistics.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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.

Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3932–3944, Online. Association for Computational Linguistics.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2020. Towards Controllable Biases in Language Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3239–3254, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5271–5277. International Joint Conferences on Artificial Intelligence Organization.

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.

Dian Yu, Kenji Sagae, and Zhou Yu. 2021. Attribute alignment: Controlling text generation from pre-trained language models. *arXiv preprint arXiv:2103.11070*.

A Limitations and Future Direction

Our framework CAT-PAW relies on token-level information, especially the BPE tokens from GPT2tokenizer. This means we have no idea of how to make decisions from a global perspective. It's hard for our framework to handle tasks such as clickbait style control that can't be summarized in bag of keywords. For future work, we will focus on controllable generation with global constraints.

Besides, our trainable regulator can outperform baseline models but is just competitive to our heuristic one. The trainable regulator is expected to possess the more powerful ability but is restricted by our easy pseudo-data creation. We may also explore a more reliable data construction method to test the boundary of its capability in the future.

B Ethical Consideration

We are fully aware that controllable generation technology has a potential to produce offensive and harmful text when maliciously used. However, it is also a powerful weapon for generating diverse contents, combating hate speech, and eliminating harmful information in pretrained language models. We believe it meaningful and beneficial for us to advance research on controllable text generation.

C Equations of Baseline Models

In detail, the decoding process is:

$$
\begin{aligned}
P(X|a) &\simeq \prod_{i=1}^{n} \left[P(x_i|x_{<i}) P(a|x_{<i})^{\lambda} \right] \\
&= \prod_{i=1}^{n} \left[\mathrm{softmax}(\mathbf{h}_i) \cdot \mathrm{softmax}(\mathbf{c}_i)^{\lambda} \right],
\end{aligned} \tag{7}
$$

where $\mathbf{c}_i$ is logits for the ith token computed by the controller $P(a|x_{<i}) = \mathrm{softmax}(\mathbf{c}_i)$. PPLM and DExperts utilize another approximation form as:

$$
P(X|a) \propto \prod_{i=1}^{n} \mathrm{softmax}(\mathbf{h}_i + \lambda\, \mathbf{c}_i). \tag{8}
$$

The main difference is that PPLM and DExperts combine output distributions of the LM and the controller before $\mathrm{softmax}(\cdot)$.

D Experiment Details

Hyperparameters are demonstrated in Table 5. PPLM's λ is composed of iteration times and step size as it provides gradient-like bias signals. Besides, we come up with a small trick for accelerating the hyperparameter tuning. We add a threshold β and get:

$$
\begin{cases}
\min\left[\lambda \times f(a, P(x_{\le i})), \beta\right], & \beta \le \lambda \\
\lambda \times \min\left[f(a, P(x_{\le i})), 1\right], & \beta > \lambda,
\end{cases}
$$

rather than $\lambda \times f(a, P(x_{\le i}))$ barely, to ensure that original methods are lower bound of ours. When weight λ is low, we can accept a more intense bias signal at the proper position. However, it's unwise to amplify the bias signal when λ is high enough.

E Additional Results

GeDi: My dog died a few weeks ago, and I recently watched this video. Not only was I deeply moved by <u>their love</u> for each other, but much like the rest of us, the grieving dogs showed the same beautiful loving behavior that makes love so...

+ T : My dog died tonight at the age of 17. She was a total joy to be with. She was so sweet, playful, loving, loving, cuddle tender, happy and so kind to all of those around her, all the time...

+ H : My dog died 2 years ago. Tallie died 2 years ago. She was 4 months old. I love her dearly and miss her so much. She is such a hardy little dog because she has a tough family life. She...

DExperts: My dog died of diabetes after nearly two decades of <u>treating my family with medication</u>, but she took to it with such enthusiasm that it touched others. She was always so thankful for life. "She brought smiles to our family," Myra said...

+ T : My dog died today. He was a lovely little husky which we only knew as an "old husky friend". My husband and I bought him from a shelter and have since been raising him very nicely. He is a very gentle one...

+ H : My dog died and you were touched for that as well. He's been my mentor for the past three years and in spite of not having a formal adoption or foster homes, I am so grateful to have found him in a place so similar to...

Figure 8: Examples on positive sentiment control.

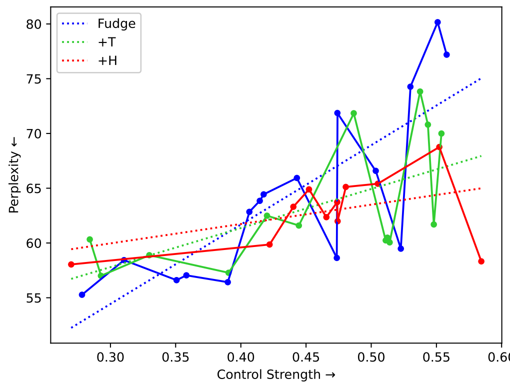

Figure 9: Trade-off between control strength and fluency of Fudge on positive sentiment control.

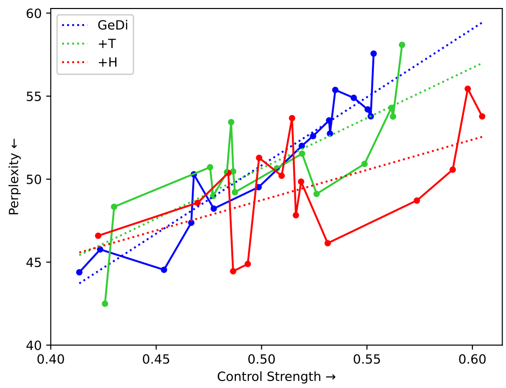

Figure 10: Trade-off between control strength and fluency of GeDi on positive sentiment control.

Model	Task	Range of λ	τ_T	τ_H	threshold β
PPLM	Positive	$[0, 3 \times 0.4]$	0.2	0.05	-
	Military	$[0, 16 \times 0.01]$	-	0.01	-
	Computers	$[0, 16 \times 0.01]$	-	0.01	-
	Legal	$[0, 16 \times 0.01]$	-	0.01	-
	Politics	$[0, 16 \times 0.01]$	-	0.005	-
	Science	$[0, 20 \times 0.01]$	-	0.005	-
	Space	$[0, 20 \times 0.01]$	-	0.005	-
	Detoxification	$[0, 3 \times 0.2]$	-	0.05	-
Fudge	Positive	$[0, 6.0]$	0.1	0.03	10.0
	Military	$[0, 10.0]$	-	0.02	12.0
	Computers	$[0, 10.0]$	-	0.015	8.0
	Legal	$[0, 3.0]$	-	0.003	6.0
	Politics	$[0, 10.0]$	-	0.001	6.0
	Science	$[0, 20.0]$	-	0.001	18.0
	Space	$[0, 20.0]$	-	0.001	17.0
GeDi	Positive	$[0, 120.0]$	0.03	0.0005	110.0
DExperts	Positive	$[0, 1.6]$	0.01	0.0006	1.3
	Detoxification	$[0, 1.6]$	-	0.05	1.3

Table 5: Hyperparameters.

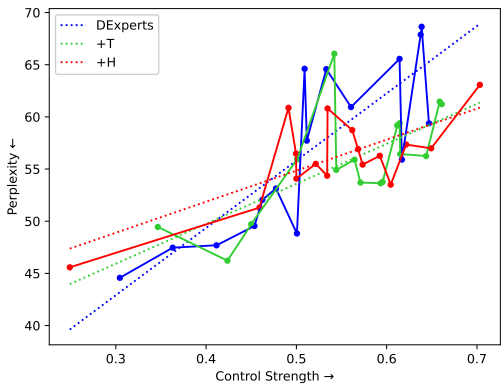

Figure 11: Trade-off between control strength and fluency of DExperts on positive sentiment control.

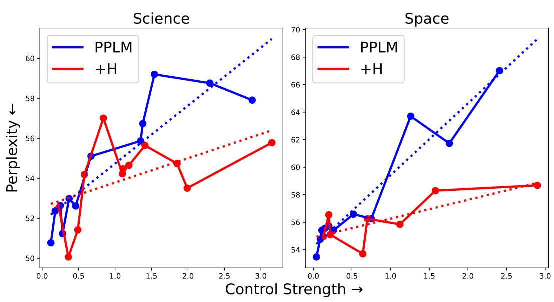

Figure 12: Trade-off between control strength and fluency of PPLM on science and space topic control.

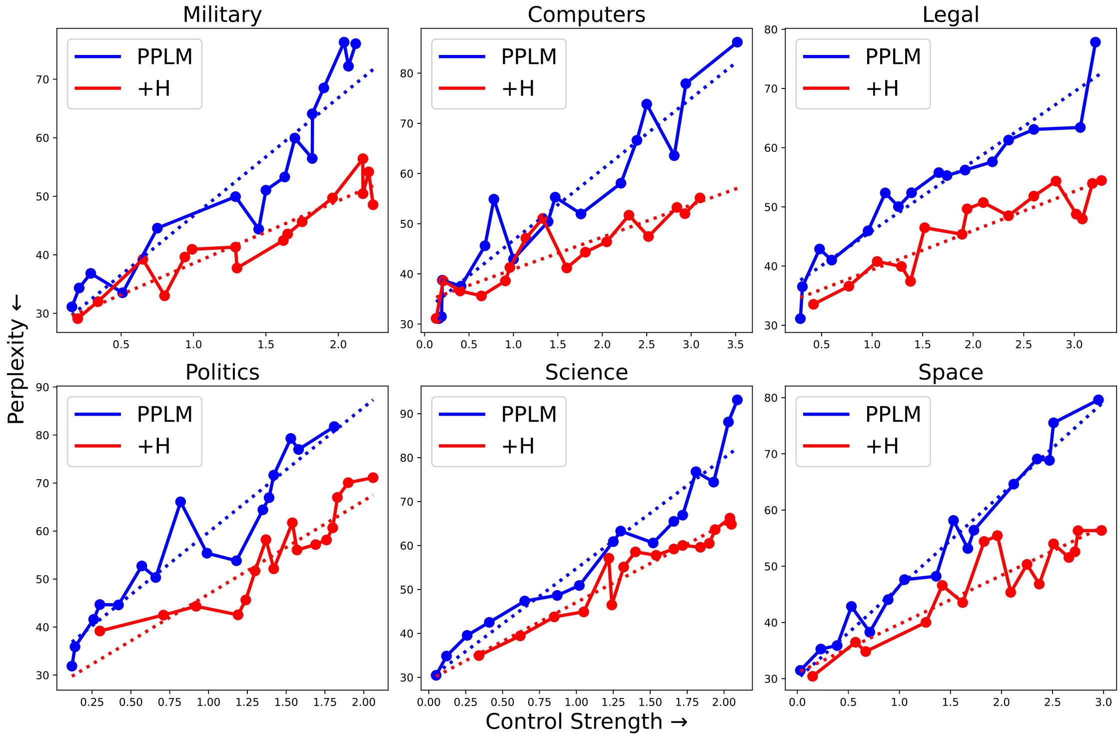

Figure 13: Trade-off between control strength and fluency of Fudge on topic control.

F Analysis on Human Evaluation

Model	Task	Kappa(%)	
		Strength	Fluency
PPLM	Positive	58.91	36.61
	Military	83.00	36.83
	Detoxification	85.00	40.83
Fudge	Positive	55.78	32.56
	Military	65.67	33.50
GeDi	Positive	58.33	38.00
DExperts	Positive	60.67	30.44
	Detoxification	84.33	40.33

Table 6: Analysis on Human Evaluation.

G Bag of Keywords for Topic Control

We use the bag of keywords collected by PPLM from www.enchantedlearning.com/wordlist.

Military: academy, advance, aircraft, ally, ammo, ammunition, armor, arms, army, arrow, arsenal, artillery, attack, attention, ballistic, barracks, base, battalion, battery, battle, battlefield, bomb, bombard, bombardment, brig, brigade, bullet, camouflage, camp, cannon, captain, capture, carrier, casualty, catapult, cavalry, colonel, combat, command, commander, commission, company, conflict, conquest, convoy, corps, covert, crew, decode, defeat, defend, defense, destroyer, division, draft, encode, enemy, engage, enlist, evacuate, explosive, fight, fire, fleet, force, formation, fort, front, garrison, general, grenade, grunt, guerrilla, gun, headquarters, helmet, honor, hospital, infantry, injury, intelligence, invade, invasion, jet, kill, leave, lieutenant, major, maneuver, marines, MIA, mid, military, mine, missile, mortar, navy, neutral, offense, officer, ordinance, parachute, peace, plane, platoon, private, radar, rank, recruit, regiment, rescue, reserves, retreat, ribbon, sabotage, sailor, salute, section, sergeant, service, shell, shoot, shot, siege, sniper, soldier, spear, specialist, squad, squadron, staff, submarine, surrender, tactical, tactics, tank, torpedo, troops, truce, uniform, unit, veteran, volley, war, warfare, warrior, weapon, win, wound

Computers: algorithm, analog, app, application, array, backup, bandwidth, binary, bit, bite, blog, blogger, bookmark, boot, broadband, browser, buffer, bug, bus, byte, cache, caps, captcha, CD, client, command, compile, compress, computer, configure, cookie, copy, CPU, dashboard, data, database, debug, delete, desktop, development, digital, disk, document, domain, dot, download, drag, dynamic, email, encrypt, encryption, enter, FAQ, file, firewall, firmware, flaming, flash, folder, font, format, frame, graphics, hack, hacker, hardware, home, host, html, icon, inbox, integer, interface, Internet, IP, iteration, Java, joystick, kernel, key, keyboard, keyword, laptop, link, Linux, logic, login, lurking, Macintosh, macro, malware, media, memory, mirror, modem, monitor, motherboard, mouse, multimedia, net, network, node, offline, online, OS, option, output, page, password, paste, path, piracy, pirate, platform, podcast, portal, print, printer, privacy, process, program, programmer, protocol, RAM, reboot, resolution, restore, ROM, root, router, runtime, save, scan, scanner, screen, screenshot, script, scroll, security, server, shell, shift, snapshot, software, spam, spreadsheet, storage, surf, syntax, table, tag, template, thread, toolbar, trash, undo, Unix, upload, URL, user, UI, username, utility, version, virtual, virus, web, website, widget, wiki, window, Windows, wireless, worm, XML, Zip

Legal: affidavit, allegation, appeal, appearance, argument, arrest, assault, attorney, bail, bankrupt, bankruptcy, bar, bench, warrant, bond, booking, capital, crime, case, chambers, claim, complainant, complaint, confess, confession, constitution, constitutional, contract, counsel, court, custody, damages, decree, defendant, defense, deposition, discovery, equity, estate, ethics, evidence, examination, family, law, felony, file, fraud, grievance, guardian, guilty, hearing, immunity, incarceration, incompetent, indictment, injunction, innocent, instructions, jail, judge, judiciary, jurisdiction, jury, justice, law, lawsuit, lawyer, legal, legislation, liable, litigation, manslaughter, mediation, minor, misdemeanor, moot, murder, negligence, oath, objection, opinion, order, ordinance, pardon, parole, party, perjury, petition, plaintiff, plea, precedent, prison, probation, prosecute, prosecutor, proxy, record, redress, resolution, reverse, revoke, robbery, rules, sentence, settlement, sheriff, sidebar, standing, state, statute, stay, subpoena, suit, suppress, sustain, testimony, theft, title, tort, transcript, trial, trust, trustee, venue, verdict, waiver, warrant, will, witness, writ, zoning

Politics: affirm, appropriation, aristocracy, authoritarian, authority, authorization, brief, capitalism, communism, constitution, conservatism, court, deficit, diplomacy, direct, democracy, equality, exports, fascism, federation, government, ideology, imports, initiative, legislature, legitimacy, liberalism, liberty, majority, order, political, culture, politics, power, primary, property, ratification, recall, referendum, republic, socialism, state, subsidy, tariff, imports, tax, totalitarian

Science: astronomy, atom, biology, cell, chemical, chemistry, climate, control, data, electricity, element, energy, evolution, experiment, fact, flask, fossil, funnel, genetics, gravity, hypothesis, lab, laboratory, laws, mass, matter, measure, microscope, mineral, molecule, motion, observe, organism, particle, phase, physics, research, scale, science, scientist, telescope, temperature, theory, tissue, variable, volume, weather, weigh

Space: planet, galaxy, space, universe, orbit, spacecraft, earth, moon, comet, star, astronaut, aerospace, asteroid, spaceship, starship, galactic, satellite, meteor