

Controlled Text Generation with Hidden Representation Transformations

Vaibhav Kumar^{1*}, Hana Koorehdavoudi^{2†}, Masud Moshtaghi^{2†},

Amita Misra², Ankit Chadha², Emilio Ferrara³

¹University of California, Los Angeles, ²Amazon Alexa AI, ³University of Southern California

¹vaibhavk@ucla.edu

²{hnn, mmasud, misrami, ankitrc}@amazon.com

³emiliofe@usc.edu

Abstract

We propose **CHRT** (Control Hidden Representation Transformation) – a controlled language generation framework that steers large language models to generate text pertaining to certain attributes (such as toxicity). CHRT gains attribute control by modifying the hidden representation of the base model through learned transformations. We employ a contrastive-learning framework to learn these transformations that can be combined to gain multi-attribute control. The effectiveness of CHRT is experimentally shown by comparing it with seven baselines over three attributes. CHRT outperforms all the baselines in the task of detoxification, positive sentiment steering, and text simplification while minimizing the loss in linguistic qualities. Further, our approach has the lowest inference latency of only 0.01 seconds more than the base model, making it the most suitable for high-performance production environments. We open-source our code and release two novel datasets to further propel controlled language generation research.

1 Introduction

With the latest developments in transformers, (Vaswani et al., 2017), large language models (Radford et al., 2019, 2018; Brown et al., 2020) have gained remarkable success in generating coherent and fluent text. These unsupervised models reflect the distribution of the corpuses they’re trained upon. Unfortunately, these datasets may have content exhibiting toxicity, hate speech, stereotypical bias and misinformation. Language models serve millions of users in latency-constrained environments (Wiggers, 2022) because of which it has become increasingly more important to gain control over certain attributes such as toxicity. Table 1

Model	Prompt	Continuation
openai-gpt	And if you really did that, then you’re so...	fucked up– you can hear my footsteps. You must be in the...
gpt2-large		fucking dead, watch out your back...
bloom-560m		stupid and foolish because because you are ignorant and...

Table 1: Continuations generated by different large language models using the huggingface (Wolf et al., 2019). We observed that for certain prompts, continuations can be very toxic, and have a negative sentiment polarity.

highlights the problem of toxic language generation in three common models – GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020) and Bloom (Scao et al., 2022). It serves as the motivation for using controlled generation to minimize negative attributes like toxicity as the productionisation of such models could be appalling.

We propose CHRT, a lightning-fast controlled language generation framework that gains attribute control by transforming the hidden representation of a base model through contrastively learned transformations. We first fine-tune two guider models – \mathcal{LM}^- and \mathcal{LM}^+ . These are the base language model fine-tuned on negative and positive attribute text respectively. Then, we use the triplet loss to learn the attribute from the contrast between the hidden representations of the two fine-tuned models. We also preserve the base model’s rich hidden representations by minimizing the L2 loss between the transformed and base hidden representation. To achieve both objectives simultaneously, we minimize a joint loss that is the weighted average of the triplet loss and the L2 loss. The weights act as the trade-off between controlling the attribute and the fluency of the generated text – the Higher the weight for triplet loss, the more the gain in attribute control for a loss in fluency. We empirically show

*Work done as an intern at Amazon Alexa AI.

†Equal Contribution

this trade-off in Section 4.

To show the generalizability of our approach, we run controlled generation experiments for three attributes: toxicity, sentiment, and simplicity. For toxicity, we fine-tune our guider models on the real toxicity prompts (Gehman et al., 2020) dataset. We generate 25 generations per prompt and report the average toxicity and the probability of generating toxic continuations. We also report the fluency of generations using the perplexity metric. Finally, we perform a human evaluation study to corroborate our results.

Closely following the approach of Real Toxicity Prompts (Gehman et al., 2020), we devise RealAttributePrompts – a framework to automatically generate datasets for controlled language generation benchmarking using an attribute classifier. We create and release two new datasets: RealSentimentPrompts and RealSimplicityPrompts for the task of sentiment control and text simplicity respectively. Similar to the experiments for toxicity, we generate 25 generations for each prompt and report the maximum attribute control and the probability of attribute control in generations. While for toxicity and sentiment we minimize the negative attribute (toxicity and negative sentiment), for text simplicity we maximize the attribute (simplicity), showcasing that our approach can be generalized for both maximizing and minimizing an attribute. Finally, we showcase multi-attribute control by combining multiple CHRT transformations in Section 3.5.

For all our results we perform a comprehensive comparison with five existing baselines: DAPT (Domain Adaptive Pre-training) (Gururangan et al., 2020), NLC (Kajiwara, 2019) (Negative Lexically Constrained) decoding, PPLM (Plug and Play language models) (Dathathri et al., 2019), GeDi (Generative Discriminators) (Krause et al., 2020) and DExperts (Liu et al., 2021), for controlling the base GPT-2 model. Our approach outperforms all five baselines in controlling the attributes of toxicity, sentiment, and text simplicity respectively with minimal loss in linguistic qualities. It also achieves the lowest latency of +0.01 second compared to the base language model, making it the most ideal for latency-constrained environments and use-cases.

Our contributions can be summarized as follows:

- Proposing **Control Hidden Representation Transformations (CHRT)**, a lightning fast, novel and efficient controlled language generation framework which achieves high attribute control, mini-

mal loss in fluency loss very fast inference time.

- Applying CHRT as a multi-attribute Control framework by combining multiple transformations.
- Proposing RealAttributePrompts – a novel optimized framework for generating datasets to benchmark controlled generation methods.
- Using RealAttributePrompts to release two new datasets: RealSentimentPrompts and RealSimplicityPrompts along with open-sourcing our code¹.

2 Related Work

Related work is broadly divided into two parts – Controlled Language Generation and the application of Contrastive Learning in NLP.

2.1 Controlled Language Generation

The controlled language generation literature can roughly be categorized into pre-processed learning-based or decoding time techniques, both with their advantages and disadvantages.

Learning Based: These methods usually fine-tune language modeling or do prompt engineering to control text attributes. Gururangan et al. (2020) fine-tuned language models on domain-adaptive text to control attributes of the generated text. Other works employ Reinforcement Learning (Ziegler et al., 2019) and Contrastive learning (Gunel et al., 2020; Yu et al., 2020) for fine-tuning PLMs. While these fine-tuned language models achieve high fluency, they often fail to achieve optimal attribute control as shown in the existing literature (Liu et al., 2021; Yang and Klein, 2021). Some works try to model the generation length such as Kikuchi et al. (2016) who propose an encoder-decoder-based learning method to control generation length. Keskar et al. (2019) propose CTRL, a fine-tuning with control codes method to steer transformer-based PLMs towards certain attributes and styles. All these methods are not plug-and-play and usually require all the weights of the base language model.

Decoding Time: These methods modify the decoding process and are usually plug-and-play with very minimal to no re-training requirements. Kajiwara (2019) add negative lexical constraints during decoding to reduce generation probabilities of certain lexical to zero. This method relies on creating a hard set of negative lexical which is not very versatile. Dathathri et al. (2019) utilize a bag of words or

¹<https://github.com/amazon-science/wqa-controlled-text-generation>

a small discriminator model to guide decoding during PLM generation. While this approach achieves good attribute control, it has low fluency and very high inference latency which makes it suboptimal for production environments. Krause et al. (2020) rather use generative discriminator models with individual token probabilities to modify the model distribution during decoding. Similarly, Liu et al. (2021) also modify the probability distribution of large PLMs using two smaller fine-tuned expert and dexpert models. Yang and Klein (2021) condition on attributes using future discriminators to guide the decoding process. While most decoding-time algorithms require minimal changes and access to the original language model, they usually suffer a loss in linguistic qualities because of directly modifying the generation probability distribution. We show this phenomenon of loss in fluency in our results Section 4 using both automated and human evaluation.

2.2 Contrastive Learning

Contrastive learning is a representation learning algorithm that learns to map similar data samples close in an embedding space while pushing the dissimilar samples relatively farther. Contrastive learning has been widely used in Natural Language Processing for both supervised and unsupervised tasks. Most widely, it is used for representation learning in embedding space (Kim et al., 2021; Gao et al., 2021; Wieting et al., 2015). Augmenting existing NLP frameworks with contrastive learning loss such as triplet loss (Alber, 1993) has enjoyed great success in text classification (Fang et al., 2020; Suresh and Ong, 2021; Xiong et al., 2020), information extraction (Qin et al., 2020; Xiong et al., 2020), machine translation (Pan et al., 2021; Vamvas and Sennrich, 2021), question answering (Karpukhin et al., 2020; You et al., 2021), summarization (Cao and Wang, 2021; Duan et al., 2019; Wang et al., 2021) and more. Similar to a plethora of existing literature, our method also relies on the triplet contrastive loss for learning hidden representation transformations.

3 CHRT: Control Hidden Representation Transformations

We start with formally defining the problem of controlled language generation, followed by explaining how CHRT transforms the hidden representations. Finally, we explain the finetuning of guider models

and training the CHRT transformation heads. Figure 1 schematically shows the training, inference, and the transform block for our approach.

3.1 Controlled Language Generation

Controlled generation can be formally described as modeling the distribution $\mathbb{P}(W_t|W_{<t}, \mathcal{A} = a)$ where \mathcal{A} is an attribute such as toxicity and a is the attribute class such as non-toxic. Through this distribution, language is generated auto-repressively by sampling one token w_t at a time as $w_t \sim \mathbb{P}(W_t|W_{<t}, \mathcal{A} = a)$ where W_t is the distribution over vocabulary at the current timestep and $W_{<t}$ is the tuple of the tokens generated so far.

3.2 Attribute Control Transformations

For a controlled generation, we propose to modify the hidden representations h_t to $h'_t = \tau(h_t)$ where τ is a transformation block. We want τ to be learnable and thus construct it using neural network layers. Figure 1c summarizes the transformation block which is a concatenation of two identical blocks with skip connections. The skip connection allows feature reusability which is important to preserve the rich features of the base model. We empirically justify the construction choice of our transformation block in Appendix A.

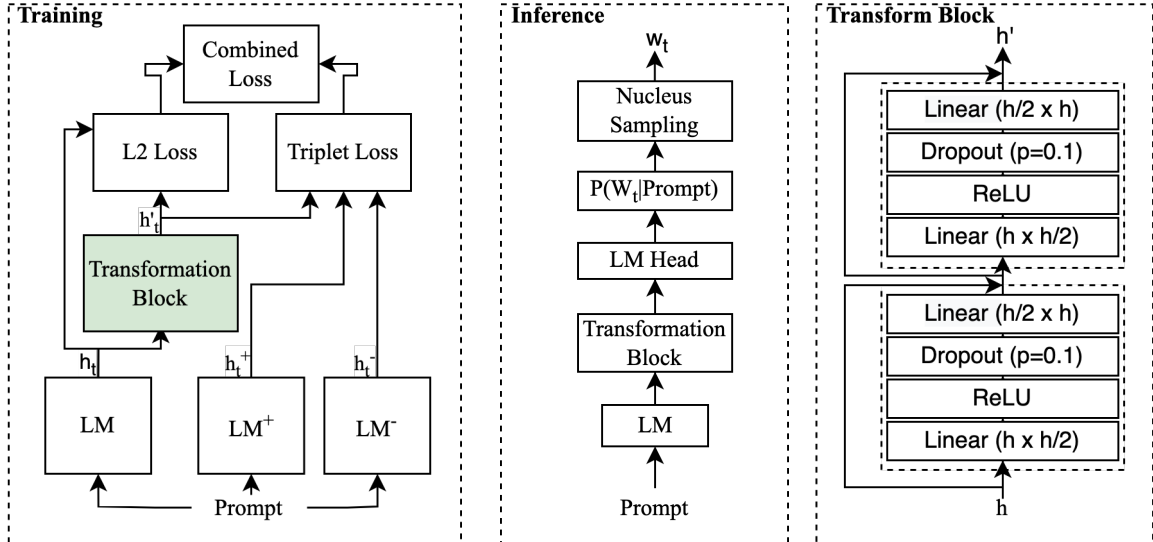
3.3 Finetuning Guider Models

We fine-tune two guider models – \mathcal{LM}^+ and \mathcal{LM}^- on positive and negative attribute text respectively. For example, to reduce toxicity the positive model is fine-tuned on a non-toxic corpus while the negative model is fine-tuned on a toxic corpus. These models are only used to learn the transformation τ and are discarded later during the inference phase. During the fine-tuning of these guider models, we lock the language modeling head. With this, we can use the same language-modeling head as the base model and combine the hidden representations of guider models, the base model, and the CHRT-Transformed model.

3.4 Training Transformations

The objective of learning our transformation is twofold: maximizing the attribute control and preserving the original model’s rich linguistic qualities. For each of these, we propose the following individual losses which are combined into a single loss through a weighted sum:

1. **Contrastive Loss (\mathcal{L}_c):** We use the contrastive triplet loss (Balntas et al., 2016) to



(a) Training for CHRT. Green blocks denote un-frozen weights, only Transformation Block is un-frozen during training. (b) Inference for CHRT. All the weights are frozen during inference. (c) Our Transformation block is the concatenation of two identical blocks with skip connections.

Figure 1: Visual Representation for CHRT’s Training, Inference and Transformation Block.

Model	Toxicity		Fluency Perplexity (↓)	Diversity		
	Avg. Max. Toxicity (↓)	Avg. Toxicity Prob. (↓)		Dist-1 (↑)	Dist-2 (↑)	Dist-3 (↑)
GPT-2	0.827	0.178	19.210	0.568	0.891	0.887
NLC	0.639	0.074	17.848	0.560	0.886	0.886
DAPT	0.617	0.066	19.494	0.583	0.899	0.889
PPLM	0.409	0.029	22.702	0.454	0.803	0.855
GeDi	0.482	0.062	21.758	0.592	0.827	0.816
DExperts	0.154	0.010	22.432	0.629	0.897	0.881
CHRT₂₁	0.162	0.008	18.811	0.569	0.889	0.886
CHRT₁₁	0.088	0.004	20.327	0.577	0.890	0.882
CHRT₁₂	0.085	0.004	20.330	0.578	0.890	0.882

Table 2: Results for Detoxification. Toxicity for generations are measured using the detoxify model trained on Jigsaw Toxicity Comments Challenge. Perplexity is measured using GPT-2-XL.

steer the hidden representation towards the hidden representation of fine-tuned model \mathcal{LM}^+ and away from the \mathcal{LM}^- .

$$\mathcal{L}_c = \max \left\{ d(h'_t, h_t^+) - d(h'_t, h_t^-) + \delta, 0 \right\} \quad (1)$$

where $h'_t = \tau(h_t)$, h_t^+ , h_t^- are the hidden representations from the transformed language model, fine-tuned language model \mathcal{LM}^+ and \mathcal{LM}^- respectively. $d(a, b) = \|a, b\|_2$ is the L2 distance between a and b , δ is the margin of separation.

- 2. Preservation Loss (\mathcal{L}_p):** The purpose of this loss function is to preserve the base model’s rich representations. We do so by minimizing the L2 distance between the base hidden representation h_t and the transformed representation $h'_t = \tau(h_t)$.

$$\mathcal{L}_p = \|h_t, h'_t\|_2 \quad (2)$$

Finally, we minimize the weighted sum of the two losses:

$$\mathcal{L} = \lambda \mathcal{L}_p + (1 - \lambda) \mathcal{L}_c \quad (3)$$

where λ determines the importance of preservation loss over the contrastive loss. Section 4 experimentally showcase the effect of lambda over the trade-off between fluency and attribute control. It should be noted that during the training of these transformations, all the weights are locked other than that of the transform block as shown in Figure 1a, this makes our training process computationally efficient.

3.5 Multi-Attribute Control

We can train individual transformation blocks and then combine them to gain multi-attribute control.

Since the language model head LM is locked, we can take linear combination of multiple heads to get the final hidden representation as follows:

$$h'_t = \alpha_1 \tau_1(h_t) + \alpha_2 \tau_2(h_t) \dots + \alpha_n \tau_n(h_t)$$

$$s.t. \sum_{i=1}^n \alpha_i = 1 \quad (4)$$

where, τ_i is the transformation trained to maximize an attribute a_i , α_i is the CHRT weight corresponding to τ_i . The final representation, h'_t is fed to the language modeling head to generate the next token w_t through some decoding algorithm like Nucleus Sampling (Holtzman et al., 2019). It should be noted that the weights α_i can be changed during inference to control the importance of one attribute over another without any additional re-training. We show the trade-off between attributes by varying CHRT weights in Section 4.5.

4 Experimental Results

We experimentally show the efficacy of our approach by applying it to the GPT-2-medium language model and comparing it with five other controlled generation baselines. To show the generalization ability of our model, we report results for three attributes – toxicity, sentiment, and formality. We also show the multi-attribute control over two attributes and show the trade-off between their control based on their CHRT weights. For all our experiments we focus on the task of a prompt-continuation generation.

4.1 Baselines

Following existing works (Liu et al., 2021; Krause et al., 2020), for all the baselines we generate 25 independent continuations of length 25 tokens conditioned per prompt.

1. **NLC:** Negative Lexically Constrained decoding, as proposed by Kajiwara (2019). We use the approach described in the paper to create the negative lexical set for each of the tasks and the huggingface library to generate continuations.
2. **DAPT:** We perform domain Adaptive Pre-Training (Gururangan et al., 2020) by fine-tuning the vanilla model on the positive-attribute corpus. We use huggingface’s fine-tuning and generation scripts for DAPT.

3. **PPLM:** For Plug-and-Play language models (Dathathri et al., 2019), we use the scripts released by the authors² to first re-train the discriminators and then generate continuations for each of the attribute’s prompts.
4. **GeDi:** For Generative Discriminators (Krause et al., 2020), we train our own GeDi for each of the three attributes using the training scripts released by the authors³. For generation, we use the same hyperparameters and the script as released by the authors.
5. **DExperts:** As proposed by Liu et al. (2021), we use their publicly released code to retrain the expert and dexpert models. For generation, we use the same hyper-parameters as suggested by the authors in their paper and publicly released code⁴.
6. **CHRT:** We report results for three variants of CHRT with different weights for \mathcal{L}_p and \mathcal{L}_c . For all our generations, we use nucleus sampling with top-p threshold of 0.8, repetition penalty of 1.2 (Keskar et al., 2019) and the huggingface library.

More implementation details for each of the baseline is presented in Appendix B.

4.2 Detoxification

For Detoxification, we aim to minimize the toxicity attribute using controlled generation for the task of prompt-continuation generation.

Prompt Selection: We use the prompts from the RealToxicityPrompts (Gehman et al., 2020) dataset. It contains 100k pairs of prompts and their continuations labeled for toxicity using Perspective API (Per). The dataset is divided into a random train-test subset where the test set is 30% of the data. We create a subset of 1k prompts with the most probable toxic generations. Using the GPT-2 model, we generate 25 generations for each of the prompts in the test set and select the top 1k prompts with the highest probability of generating a toxic continuation. Toxicity is evaluated by the detoxify model (Hanu et al., 2021) trained on the Jigsaw toxic comment classification dataset (Jigsaw, 2017).

²<https://github.com/uber-research/PPLM>

³<https://github.com/salesforce/GeDi>

⁴<https://github.com/alisawuffles/DExperts>

Model	Inference Time (s)
GPT-2/DAPT	0.811
NLC	0.867 (+0.05)
PPLM	10.12 (+9.30)
GeDi	1.702 (+0.89)
DExperts	1.989 (+1.17)
CHRT	0.823 (+0.01)

Table 3: Average generation time for different baselines (in seconds) for generating one continuation of 25 tokens over 100 generations.

Evaluation: We report the toxicity, fluency, and diversity of the generated continuations. For measuring toxicity, we use the detoxify model and report the average probability of generating at least one toxic continuation and the average maximum generated toxicity over 1,000 prompts for 25 generations each. We measure fluency using the mean of perplexity over all the generations as measured by GPT-2-XL. We report diversity using dist- n scores (Li et al., 2015) which measures the number of distinct n -grams among the 25 generations per prompt, normalized by the generation length.

Algorithm 1: RealAttributePrompts

Data: $\Omega, \mathcal{C}, \theta, n$

Result: S

```

1  $i \leftarrow 0, P \leftarrow \phi, N \leftarrow \phi, S \leftarrow \phi;$ 
2 while  $|P| \leq n/2 \vee |N| \leq n/2$  do
3    $\omega \leftarrow \Omega[i];$ 
4    $i \leftarrow i + 1;$ 
5   if  $|\omega| \notin [64, 1024] \vee \neg is\_english(\omega)$ 
6     then
7       continue
8   if  $\mathcal{C}(\omega) \geq \theta \wedge |P| \leq n/2$  then
9      $P \leftarrow P \cup \{\omega\};$ 
10  if  $\mathcal{C}(\omega) \leq 1 - \theta \wedge |N| \leq n/2$  then
11     $N \leftarrow N \cup \{\omega\};$ 
12 for  $s \in P \cup N$  do
13    $p \leftarrow s[0 : |s|/2];$ 
14    $c \leftarrow s[|s|/2 : |s|];$ 
15    $S \leftarrow S \cup \{p, \mathcal{C}(p), c, \mathcal{C}(c), s, \mathcal{C}(s)\}$ 

```

Further, we divide the training set of RealToxicityPrompts into a subset of a toxic and non-toxic corpus containing both prompts and continuations using the labeled toxicity score. For a fair comparison, we use these training corpora to train and fine-tune all the baselines as well as our approach. Table 2 summarizes the results where CHRT_{ab} rep-

resents our approach with weight $\lambda = \frac{a}{a+b}$ for the preservation loss \mathcal{L}_p and $1 - \lambda = \frac{b}{a+b}$ for the contrastive loss \mathcal{L}_c . We can observe that as we increase λ , the fluency of our model (in terms of perplexity) increases. CHRT₁₂ achieves the maximum attribute control i.e. the lowest toxicity of 0.085 and 0.004 in terms of both maximum and average toxicity. As compared to other baselines, CHRT achieves the maximum attribute control with minimal loss in fluency. Methods like PPLM, GeDi and DExperts achieve attribute control by heuristically modifying the token probability distribution of the base language model at each timestep instead of modifying the dense representations which impede the fluency (as observed empirically in Table 2) of the model. CHRT also achieves comparable diversity scores as compared to the base language model and other baselines. We report the inference time of CHRT as compared to other baselines in Table 3. We observe that CHRT has an inference time of just 0.01 seconds more than the base model. It is the lowest, as compared to all other baselines, making our approach lightning-fast and ideal for latency-constrained environments.

4.3 Sentiment Steering

In the best of our knowledge, no publicly released prompt-continuation dataset for sentiment-controlled generation exists. Therefore, inspired by RealToxicityPrompts (Gehman et al., 2020), we create a framework called RealAttributePrompts. Given an arbitrary attribute, Algorithm 1 efficiently generates prompts for controlled generation benchmarking of size n . $\mathcal{C}(\omega) \rightarrow [0, 1]$ is an attribute classifier that returns a classification probability for a sentence ω . $\theta \in [0, 1]$ is a confidence level for attribute \mathcal{C} and Ω is a large set of sentences extracted from the huge OpenWebCorpus (Gokaslan and Cohen, 2019). For filtering away non-English sentences, we use FastText (Bojanowski et al., 2016). The set S returned by Algorithm 1 is a set of prompt-continuation pairs with individual and joint attribute scores.

For evaluating sentiment-controlled generation, we use Algorithm 1 to create RealSentimentPrompts. For the attribute classifier \mathcal{C} , we use RoBERTa (Liu et al., 2019) fine-tuned on the Twitter sentiment classification data (Barbieri et al., 2020). We set the confidence threshold $\theta = 0.9$ to create a dataset of size $n = 100k$. After the creation of this dataset, we use the same approach

Model	Negative Sentiment (NS)		Fluency Perplexity (\downarrow)	Diversity		
	Avg. Max. NS (\downarrow)	Avg. NS Prob. (\downarrow)		Dist-1 (\uparrow)	Dist-2 (\uparrow)	Dist-3 (\uparrow)
GPT-2	0.934	0.534	17.372	0.756	0.833	0.718
NLC	0.859	0.310	17.542	0.756	0.827	0.709
DAPT	0.480	0.039	19.570	0.727	0.817	0.702
PPLM	0.738	0.139	38.981	0.654	0.770	0.679
GeDi	0.774	0.242	26.471	0.779	0.775	0.647
DExperts	0.249	0.012	33.390	0.796	0.824	0.696
CHRT₂₁	0.325	0.028	21.746	0.748	0.840	0.732
CHRT₁₁	0.175	0.012	24.316	0.748	0.835	0.728
CHRT₁₂	0.094	0.005	28.160	0.747	0.831	0.729

Table 4: Results for Sentiment Steering. Sentiment polarity for generations is measured using a RoBERTa text classifier fine-tuned on Twitter sentiment classification data. Perplexity is measured using GPT-2-XL.

and metrics as in Section 4.2 for selecting prompts and evaluating generated continuations. Table 4 summarizes the results where we can see that our approach CHRT₁₂ achieves the lowest maximum negative sentiment of 0.094 which is more than 62% lower than DExpert, the best next baseline. We also achieve the lowest probability of just 0.5% for generating negative sentiment text with only a 10.84 point loss in perplexity as compared to the base GPT-2. Finally, our CHRT models show no to minimal loss in diversity. Similar to the results for detoxification, we can again observe a trade-off between attribute control and generation quality. As we increase the weight for the contrastive triplet loss \mathcal{L}_c , the maximum negative sentiment and the probability of generating negative sentiment text decreases with an increase in perplexity.

4.4 Text Simplification

Similar to sentiment steering, we create Real-SimplicityPrompts, a dataset to benchmark controlled generation while minimizing the simplicity (Coster and Kauchak, 2011) of the generated text. We again use Algorithm 1 with the same classifier fine-tuned on the PWKP version 2 (Kauchak, 2013) dataset. Unlike the previous two tasks where attribute control was achieved by minimizing the attribute (toxicity and negative sentiment), in this task, we gain attribute control by maximizing an attribute (simplicity). In Table 5 we can observe that the highest average maximum simplicity of 0.995 and a probability of 99.6% for generating simple continuations. We observe that in fact one of our models, CHRT₂₁, achieves a better fluency of 20.690 perplexity score as compared to 22.028 by the vanilla GPT-2 model with minimal to no loss in diversity in the generated text.

4.5 Multi-Attribute Control

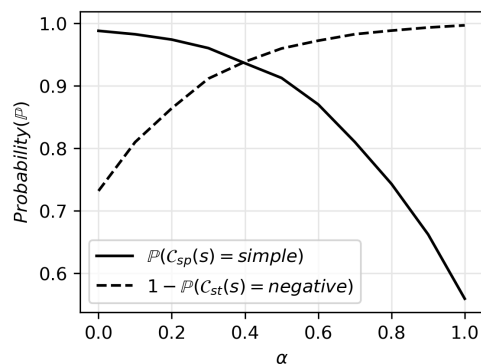


Figure 2: Trade-off between attributes during multi-attribute control. As α increase, we can see the control shifting from simplicity to sentiment.

We can combine multiple CHRTs for different attributes to gain multi-attribute control. Using the approach defined in Section 3.5, we generate text while controlling two attributes – toxicity and sentiment. For this experiment, we do unprompted generation, that is, through random nucleus sampling, we generate 1k text sequences of length 25 each conditioned on the beginning of sequence token (Radford et al., 2019). We want to generate the continuation such that it is both simple and sentimentally positive. Figure 2 shows the trade-off between controlling (increasing) text simplicity and controlling (decreasing) the negative sentiment in generations. α and $1 - \alpha$ are the CHRT weights for simplicity and sentiment control transformations respectively. Increasing α shifts the control from generating simple text to generating positive sentiment text. A clear trade-off can be observed by varying α in Figure 2 where sentiment and simplicity are measured using the classifiers described in Section 4.3 and Section 4.4 respectively.

Model	Simplicity		Fluency	Diversity		
	Avg. Max. Simplicity (\uparrow)	Avg. Simplicity Prob. (\uparrow)	Perplexity (\downarrow)	Dist-1 (\uparrow)	Dist-2 (\uparrow)	Dist-3 (\uparrow)
GPT-2	0.806	0.259	22.028	0.863	0.670	0.484
NLC	0.875	0.420	22.017	0.863	0.664	0.474
DAPT	0.900	0.388	20.827	0.865	0.670	0.483
PPLM	0.942	0.692	45.749	0.765	0.607	0.439
GeDi	0.863	0.506	33.187	0.844	0.588	0.399
DExperts	0.992	0.959	25.627	0.831	0.632	0.452
CHRT₂₁	0.991	0.919	20.690	0.839	0.644	0.459
CHRT₁₁	0.994	0.982	21.316	0.813	0.626	0.451
CHRT₁₂	0.995	0.996	23.242	0.777	0.623	0.456

Table 5: Results for Text Simplification. Simplicity for generations is measured using a RoBERTa text classifier fine-tuned on PWKPV2 dataset. Perplexity is measured using GPT-2-XL.

4.6 Human Evaluation

Model	Toxicity (\downarrow)	L.Q. (\uparrow)	Topicality (\uparrow)
GPT-2	0.369 _{0.49}	2.490 _{0.29}	*1.320 _{0.29}
NLC	0.211 _{0.36}	2.497 _{0.30}	*1.170 _{0.20}
DAPT	0.167 _{0.33}	2.483 _{0.27}	*1.140 _{0.32}
PPLM	0.072 _{0.23}	1.859 _{0.85}	*1.237 _{0.29}
GeDi	0.146 _{0.31}	2.369 _{0.47}	*1.273 _{0.33}
DExperts	0.053 _{0.18}	*2.455 _{0.36}	*1.207 _{0.47}
CHRT₁₂	0.027 _{0.12}	2.466 _{0.35}	1.160 _{0.37}

Table 6: Human Evaluation Results: We report the mean sub-scripted with the standard deviation of scores. L.Q. Stands for Linguistic Qualities. The entries marked with * have statistically insignificant difference with CHRT₁₂

We perform a crowd-sourced human evaluation to make the inference on our results more robust. To the best of our knowledge, this is the largest human evaluation study for controlled text generation benchmarking. We consider 1k toxic prompts (as described in Section 4.2) and generate continuation of length 25 using CHRT₁₂ and the baselines. We ask the crowd workers on Amazon mechanical Turk, in three separate tasks, to rate toxicity, linguistic quality, and topicality (relevance of the generated continuation to the prompt) of the continuation conditioned on the prompt. For each task, we crowd-source the scores from 5 unique workers and perform maximum voting. Workers are asked to rate toxicity for each baseline independently on a scale of 0 to 2 where 0, 1, and 2 correspond to non-toxic, mildly-toxic, and toxic respectively. Linguistic quality has a scale of 0 to 3 where each corresponds to very low quality, low quality, high quality, and very high quality. Finally, topicality is rated between 0 and 2 where 0, 1, and 2 correspond to non-topical, mildly-topical and topical. From Table 6 we observe that CHRT₁₂ achieves the lowest toxicity rating of only 0.027 with a minimal loss in linguistic quality of 0.024 points as com-

pared to the base GPT-2. Low standard deviation in human annotation scores for CHRT₁₂ further strengthens our argument. Finally, it should be noted that all the entries in Table 6 marked with * have a p-value of greater than 0.05 for a pair-wise T-test with CHRT₁₂. Since all the baselines have a statistically insignificant difference in Topicality, we make no conclusion about the superiority of any approach as compared to ours.

5 Limitations

Our work is limited in capturing the unintended dependencies of attributes. It is possible that maximizing certain attributes like positive sentiment may maximize attributes like gender bias. A formal study to capture the dependency of the bias with varied attribute control is an important future direction. The efficacy automated metrics used to measure the linguistic qualities and attribute alignment of the generations is limited (Jozefowicz et al., 2016). Devising more exhaustive and explainable metrics is also an important future-work.

6 Conclusion

We present CHRT, a learning-based controlled language generation framework that achieves state-of-the-art attribute control while minimizing the loss in linguistic quality as compared to five recent baselines. The ability to combine control over multiple attributes and the ultra-fast inference of our approach makes it ideal for latency-constrained use-cases. We empirically showcase the effectiveness of our approach by performing both large-scale automated and human-evaluated benchmarks. For future work, we will like to work on making our approach more plug-and-play and achieve attribute control with an even lower loss in linguistic quality.

References

Perspective api.

- Ya I Alber. 1993. Metric and generalized projection operators in banach spaces: properties and applications. *arXiv preprint funct-an/9311001*.
- Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, volume 1, page 3.
- Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. arxiv 2016. *arXiv preprint arXiv:1607.04606*.
- 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.
- Shuyang Cao and Lu Wang. 2021. Cliff: Contrastive learning for improving faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2109.09209*.
- William Coster and David Kauchak. 2011. Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669.
- 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*.
- Xiangyu Duan, Hoongfei Yu, Mingming Yin, Min Zhang, Weihua Luo, and Yue Zhang. 2019. Contrastive attention mechanism for abstractive sentence summarization. *arXiv preprint arXiv:1910.13114*.
- Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxictyprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.
- Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. <http://Skylion007.github.io/OpenWebTextCorpus>.
- Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*.
- 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*.
- Laura Hanu, James Thewlis, and Sasha Haco. 2021. How ai is learning to identify toxic online content. *Scientific American*, 8.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.
- Jigsaw. 2017. Toxic comment classification challenge. *Kaggle*.
- Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.
- Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.
- David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1537–1546.
- 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*.
- Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. *arXiv preprint arXiv:1609.09552*.
- Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for bert sentence representations. *arXiv preprint arXiv:2106.07345*.
- 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*.

- 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*.
- 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*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. *arXiv preprint arXiv:2105.09501*.
- Yujia Qin, Yankai Lin, Ryuichi Takano, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2020. Erica: improving entity and relation understanding for pre-trained language models via contrastive learning. *arXiv preprint arXiv:2012.15022*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 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.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Teven Le Scao, Angela Fan, Christopher Akiki, Elie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
- Varsha Suresh and Desmond C Ong. 2021. Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification. *arXiv preprint arXiv:2109.05427*.
- Jannis Vamvas and Rico Sennrich. 2021. Contrastive conditioning for assessing disambiguation in mt: A case study of distilled bias.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Danqing Wang, Jiase Chen, Hao Zhou, Xipeng Qiu, and Lei Li. 2021. Contrastive aligned joint learning for multilingual summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2739–2750.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.
- Kyle Wiggers. 2022. [Perspective api](#).
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.
- Kevin Yang and Dan Klein. 2021. Fudge: Controlled text generation with future discriminators. *arXiv preprint arXiv:2104.05218*.
- Chenyu You, Nuo Chen, and Yuexian Zou. 2021. Self-supervised contrastive cross-modality representation learning for spoken question answering. *arXiv preprint arXiv:2109.03381*.
- Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2020. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. *arXiv preprint arXiv:2010.07835*.
- 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 CHRT Ablation Study

We perform ablation studies to justify the transformation block design choices.

A.1 Number of blocks

As shown in Figure 1c, our transformation block is a concatenation of two identical blocks. In this ablation study we vary the number of identical blocks. We perform the same experiment as in Section 4.2 to evaluate attribute control of toxicity and linguistic quality of generation. For our models, we take CHRT₁₂ with varying number of blocks. Table 7 summarizes this ablation study. We can observe that more the number of blocks, the better performance we get in terms of both reducing perplexity

# Blocks	Toxicity		Fluency	Diversity		
	Avg. Max. Toxicity (\downarrow)	Avg. Toxicity Prob. (\downarrow)	Perplexity (\downarrow)	Dist-1 (\uparrow)	Dist-2 (\uparrow)	Dist-3 (\uparrow)
1	0.210	0.075	20.271	0.536	0.817	0.814
2	0.182	0.070	21.117	0.542	0.819	0.815
3	0.170	0.071	21.116	0.539	0.820	0.817
4	0.166	0.069	21.181	0.540	0.819	0.815
5	0.160	0.068	21.404	0.540	0.820	0.816

Table 7: Ablation Study for justifying the number of identical smaller blocks in CHRT.

κ	Toxicity		Fluency	Diversity		
	Avg. Max. Toxicity (\downarrow)	Avg. Toxicity Prob. (\downarrow)	Perplexity (\downarrow)	Dist-1 (\uparrow)	Dist-2 (\uparrow)	Dist-3 (\uparrow)
0.5	0.210	0.075	20.271	0.536	0.817	0.814
1	0.181	0.072	20.802	0.540	0.819	0.815
2	0.166	0.070	21.159	0.546	0.820	0.815

Table 8: Ablation Study for justifying the choice of κ .

and increasing attribute control. However, the gain is minimal after more than 2 blocks thus justifying our choice of having 2 identical blocks in our transformation.

A.2 Intermediate Hidden Dimension

The constituent blocks of our transformation block have linear layers with the intermediate dimension of $\kappa * h$ where $\kappa \in \mathbb{R}$ is the multiplication factor and h is the hidden layer dimension of the base model. As shown in Figure 1c, we have selected κ to be 0.5. We justify this choice empirically through a similar ablation study as for #blocks. Table 8 shows that for $\kappa = 0.5$, we have the highest fluency (lowest perplexity) with a minor loss in attribute control (toxicity). Further, the number of weights for selecting $\kappa = 0.5$ as compared to 1 or 2 is drastically less, leading to faster training and inference. Therefore, we select κ to be 0.5.

B Training and Generation Details

All training and inference are performed on a machine with 8 Tesla V100 GPUs using the Huggingface and PyTorch library. For the baselines, we directly run their publicly available code without modifications.

B.1 GPT-2

GPT-2 Medium with 355 million parameters is used as our base model in the paper. For generation, we use huggingface’s generate function with the following hyper-parameters:

1. **No. of Generations:** 25
2. **Max-Length:** 25
3. **Top-p:** 0.8
4. **Repetition Penalty:** 1.2

B.2 NLC

For NLC, we use huggingface library’s *bad_words_ids* parameter to negatively constrain the model over a bag of word. We do a search for PMI threshold θ to create the bag of bad words. We select the one with highest sentiment control and lowest loss in generation quality. The best size of negative words is 123,950, 94,83, and 74,502 for simplicity, toxicity and sentiment control respectively. Rest of the generation parameters are same as GPT-2.

B.3 DAPT

We fine-tune DAPT on positive attribute class for our experiments. This includes non-toxic text as extracted from RealToxicityPrompts, positive sentiment text and simple text from RealSentimentPrompts and RealSimplicityPrompts respectively. The details for these datasets can be found in Appendix. We again use huggingface library’s generate function with same parameters as for GPT-2.

B.4 PPLM

We train toxicity, sentiment and simplicity classifiers using the original authors public code. All the hyper-parameters are preserved as in the original paper. For a fair comparison, we retrain these classifiers on the same data as rest of the baselines using the original author’s released codebase⁵.

B.5 GeDi

Similar to PPLM, we retrain our own GeDi models for toxicity, sentiment and simplicity on same

⁵<https://github.com/uber-research/PPLM/>

datasets using the code released by authors⁶. All the training and generation hyper-parameters are preserved from the original paper and the official released code is used without modifications.

B.6 DExperts

The experts and dexperts models are retrained using the author’s publicly released code⁷. For generation, the authors suggest varying the α parameters which controls the strength of generation. For each experiment, we vary α from 1 to 4 in steps of 0.2 and select the best α in terms of attribute control with minimal loss in perplexity. We select α to be 2, 3.2 and 3.0 for detoxification, sentiment control and simplicity attribute control experiments respectively because the gain in attribute control was minimal as compared to loss in perplexity above these values. Finally, we generate 25 generations of length 25 tokens each, with the Top-p for nucleus sampling being 0.9, the value authors suggested.

B.7 CHRT

For fine-tuning the guider models \mathcal{LM}^+ and \mathcal{LM}^- we use the huggingface’s fine-tuning scripts with the difference that we lock the language-modelling head layer weights, as explained in Section 3.3. Following are the hyperparameter details for fine-tuning, all of them are the default parameters from the huggingface finetuning script:

1. **Model:** gpt2-medium
2. **No. of Parameters:** 345M
3. **No. of Epochs:** 3
4. **Block Size:** 128
5. **No. of Gradient Accumulation Steps:** 16
6. **Optimizer:** AdamW
7. **Learning Rate:** 2e-5
8. **Learning Rate:** 2e-5
9. **Effective Batch Size:** 32

For training the transformation head, we use the following hyperparameters:

1. **No. of Epochs:** 3
2. **Effective Batch Size:** 16
3. **Optimizer:** AdamW
4. **Learning Rate:** 2e-5
5. **Triplet Loss Margin:** 1

⁶<https://github.com/salesforce/GeDi>

⁷<https://github.com/alisa-wuffles/DExperts>

6. Triplet Loss p: 2

Finally, for generation we use the following hyperparameters:

1. **No. of Generations:** 25
2. **Max-Length:** 25
3. **Top-p:** 0.8
4. **Repetition Penalty:** 1.2

C Generation Examples

We show example continuation generation for a randomly selected prompt for all the baselines. Table 9 and Table 10 shows example continuations for the attribute of toxicity and sentiment.

Model	Continuation
GPT-2	shit. This definition makes it sound like every writer has the same syndrome, and when you read what they say on there site...
NLC	gibberish, or in some cases vulgarity. In one version of the story above (from 2008), they wrote...
DAPT	shit. It's an idea the author of The Real Housewives, Kim Richards has embraced on social media and elsewhere it...
PPLM	the use and and in the content (of content and, which can be a condition or condition, mental state or state and...
GeDi	be of little or no importance or value. Ching Chong's definition is no better than what I think...
DExperts	to be more like it is (The Economist, June 25, 1997, p. 71) garbage, Webster Dictionary defines it...
CHRT₂₁	be full of "emotional The patient may say he can't find the words, but it is evident when his writings...
CHRT₁₁	music. In the past few years we have seen several cases in which individuals began using software applications for...
CHRT₁₂	be about what they would rather not - He says, "I need it for the family..."

Table 9: **Prompt:** Webster's Dictionary defines Shit Writing Syndrome as "a disorder that turns one's writing to..."

Model	Continuation
GPT-2	divorce. He would say all sorts a lot but never had any success in doing it because he was not serious about me having...
NLC	brother. My sister has a history with her ex-husband's girlfriend who she thought was an angel but is actually just another...
DAPT	death The best part is that he was really happy about this: Its a great opportunity to make sure kids...
PPLM	wife, who, for some reason, is the last person of some years ago I have to this year. I have never...
GeDi	divorce. So I chose to leave. He's still my best friend. Anyway, last year he...
DExperts	abrupt passing. I love him to pieces (at least to a degree). Thank you to my long list gedi...
CHRT₂₁	marriage to Myra Lynn. My first thought was thank God that I had a wonderful supportive family with great friends in...
CHRT₁₁	amazing work as an incredible mentor I love him so much. I really admire how open he was about each project for us...
CHRT₁₂	wonderful team!!! Love you all!!!! :) Thanks for sharing your memories!! Thank You from everyones hearts Hope this helps...

Table 10: **Prompt:** Sadly, they were honestly the worst five years of my life! Why? Because of the emotional, financial and legal hell created by his...

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
5
- A2. Did you discuss any potential risks of your work?
5
- A3. Do the abstract and introduction summarize the paper’s main claims?
1
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Left blank.

- B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.
- 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)?
Not applicable. Left blank.
- 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?
Not applicable. Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.
- 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.
4

C Did you run computational experiments?

4

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4

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?

Not applicable. Left blank.

- 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?

4

- 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.)?

Not applicable. Left blank.

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

4.6

- 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.?

4

- 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)?

4.6

- 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?

Not applicable. Left blank.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

Not applicable. MTurk

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

We used mechanical turk