Attribute Alignment: Controlling Text Generation from Pre-trained Language Models

Dian Yu[1] dianyu@ucdavis.edu
Kenji Sagae[1] sagae@ucdavis.edu
Zhou Yu[2] zhouyu@cs.columbia.edu
[1]University of California, Davis
[2]Columbia University

Abstract

Large language models benefit from training with a large amount of unlabeled text, which gives them increasingly fluent and diverse generation capabilities. However, using these models for text generation that takes into account target attributes, such as sentiment polarity or specific topics, remains a challenge. We propose a simple and flexible method for controlling text generation by aligning disentangled attribute representations. In contrast to recent efforts on training a discriminator to perturb the token level distribution for an attribute, we use the same data to learn an alignment function to guide the pre-trained, non-controlled language model to generate texts with the target attribute without changing the original language model parameters. We evaluate our method on sentiment- and topic-controlled generation, and show large performance gains over previous methods while retaining fluency and diversity.

1 Introduction

While large pre-trained language models (LM) have advanced text generation with coherent language by training on a large amount of unlabeled data (Radford et al., 2018; Yang et al., 2019; Raffel et al., 2020), they are not controllable. For instance, given the prompt “The issue focused on”, GPT-2 (Radford et al., 2019) can generate a high-quality sentence, but it cannot take extra input such as “positive” or “business” to guide the sentence towards a positive sentiment or business-related topic, due to the lack of attribute labels during training.

To solve the discrepancy between training and inference, one direction is to train an LM from scratch with some supervision such as control codes in CTRL (Keskar et al., 2019). Nevertheless, this method requires training an LM with a large number of parameters, and is limited by the attributes used during pre-training. Another direction is to fine-tune the pre-trained LM on some annotated datasets. This usually requires updating all the parameters in the model, which incurs large computational costs with current large LMs that have hundreds of millions or billions of parameters, and may result in an LM highly relevant only to the specific training data. For example, one can fine-tune a large pre-trained LM on product reviews labeled with sentiment to generate positive and negative sentences, but the fine-tuned model will tend to generate sentences like those from product reviews. Both these methods require training all the parameters of the model. Alternatively, recent research leverages a discriminator to re-weight output distributions (Holtzman et al., 2018) or to perturb latent representations in the token level such as in PPLM (Dathathri et al., 2020) without changing the pre-trained LM. However, raising target-relevant token probabilities may lead to less fluent sentences. In addition, updating gradients at the token level makes decoding expensive and slow.

In this paper, we propose `Attribute Alignment` to infuse attribute representations into a pre-trained unconditional LM without changing the LM parameters. We are inspired by language codes which guide multilingual translation models to translate to the target language (Johnson et al., 2016). However, because attributes signals are not trained with the LM during large-scale pre-training (Johnson et al., 2016; Keskar et al., 2019), we introduce an alignment function to bridge attribute representations to the LM so that it can interpret the weights in the attribute representations.

Specifically, we encode an *attribute* (e.g. positive, negative, business, military, etc.) with a pre-trained LM and learn an alignment function to transform the attribute representation. To train the alignment function, we use the same annotated data used to train discriminators in token-level perturbation methods (Dathathri et al., 2020) so that the self-attention to the aligned attribute representation will guide the LM with a language modeling

Attribute	Generated Text
None	The issue focused on a 2008 decision by the United States Court of Appeals for the Ninth Circuit, in San Francisco, that denied local restaurants advance notice of changes to their menus, even when that change had not been submitted to ...
positive	The issue focused on returning to the simple premise that dialogue is more effective than banal reactions. They demonstrate very good personal style with establishing dialogue and bringing about a good point of view. Most fantastic of all ...
negative	The issue focused on a false belief that treatment can never be "good enough" and that long-term treatment only "cures" a person. This does not account for why this is the case: Patients with the ...
business	The issue focused on the regulations preventing banks and other entities in the financial sector from moving money across foreign borders without the consent of its investors.
athlete	The issue focused on Robinson, who went to camp with his hometown team after being released by the Seattle Seahawks, though it was ruled an emergency by the National Football League.
military	The issue focused on whether servicemen and women should be allowed to opt out of serving overseas. It was also about whether making it easier for American troops to return home would help their families.

Table 1: Examples generated using the proposed alignment function with Bayes disentanglement (ACB). Tokens underscored are the prompts. We use a classifier to select sentences (see Section 4.3.1) with the highest target attribute predication probability and present the examples here (i.e., the results are not cherry-picked). “None" indicates non-controlled generation from the original GPT-2 model. “business" is from AG News corpus, “athlete" is from DBpedia corpus, and “military" is not in the training data (zero-shot).

objective on the attribute-related dataset. In contrast to fine-tuning, this does not involve training LM parameters, thus we can do controlled text generation without sacrificing the linguistic quality of the original LM. In addition, we disentangle undesirable features from the training data using a principled approach based on Bayes' Rule. Because of the way the attributes are encoded, the end result is that the generation process can be controlled using arbitrary attributes expressed as words or phrases. Table 1 shows text generated using the prompt *The issue focused on* with various control attributes. We evaluate our proposed method on sentiment and topic control and show better performance than previous state-of-the-art methods in both controlling effectiveness and language quality.

2 Related Work

Controlled text generation To interpolate a controlling factor, concatenating the attribute to the input sequence is the most straightforward approach and has been commonly used in text and knowledge grounded generation (Dinan et al., 2019; Prabhumoye et al., 2020). Keskar et al. (2019) propose to pre-train a large conditional language model with available labels such as URLs in Reddit for large LM control. This method can be effective in conditional modeling, but requires a substantial amount of resources for pre-training and is greatly limited by the number of labels used during pre-training (e.g. 55 control codes in CTRL). Another approach is to concatenate the attribute representation to the hidden states using linear transformation (Hoang et al., 2016; Fu et al., 2018) or latent variables (Bowman et al., 2016; Wang et al., 2019). These approaches require training from scratch or fine-tuning the entire pre-trained model to incorporate the external target attributes and model conditional probability (Ficler and Goldberg, 2017). In addition, they always require carefully designed Kullback-Leibler (KL)-Divergence and adversarial training to generate out-of training domain text with the desirable attribute only (Romanov et al., 2019). In comparison, our proposed method does not require fine-tuning the original LM so that we can make use of the high quality pre-trained LM while controlling the target attributes.

Instead of fine-tuning the whole model, Houlsby et al. (2019) propose to add Adapters, which are task-specific parameters to transformer layers for each task in language understanding tasks. Different from Adapters, our method only requires learning one attribute alignment function for all the attributes to do controlled generation. An alternative is to take a pre-trained unconditional LM and perturb the hidden states towards a target attribute in a plug and play manner (Nguyen et al., 2017). PPLM proposes to train a classifier or bag-of-words to increase the likelihood of the target attribute in the hidden state before generating the next token (Dathathri et al., 2020). Similar to ours, their method does not require changing the pre-trained LM and they are able to control sentiment and various topics. However, ascending conditional probability in the token level to shift the distribution towards target-related tokens can lead to de-

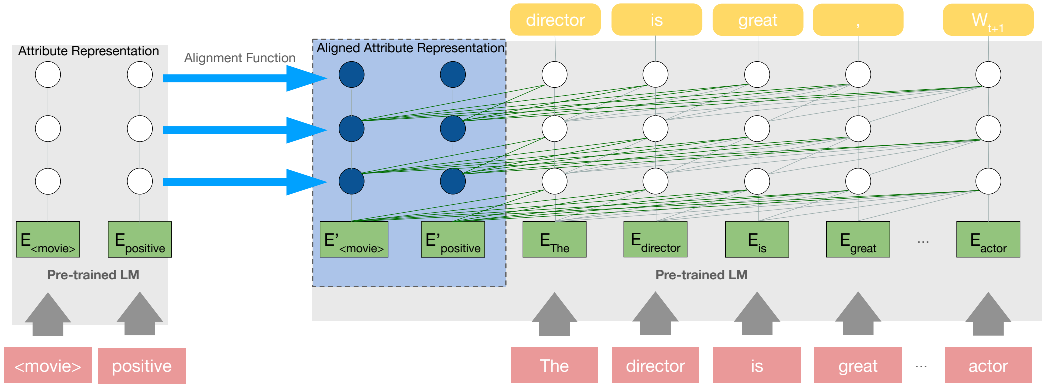

Figure 1: Attribute Alignment model architecture with corpus representation disentanglement. We train the alignment function (an MLP in our experiment shown as blue arrows) to transform attribute (e.g. `positive sentiment`) representation (encoder hidden states in the left grey box) to aligned attribute representation (blue shade box in the middle). The training objective is to generate attribute-related sentences in the training dataset by attending to aligned attribute representation (green lines) in addition to regular self-attention (grey lines).

generation (Holtzman et al., 2020) and is slow at inference time. In contrast, our method learns an alignment function on hidden representations of the attribute so that tokens can do self-attention with the attribute without breaking the pre-trained self-attention in the LM. During generation, we can simply send the attribute as a signal for conditional generation. Our method is uniform for different attributes such as sentiment and topics, and is more efficient and flexible.

Attribute representation learning Liu and Lapata (2018) split hidden representations to encourage different dimensions to learn different attributes for document representation. In comparison, Romanov et al. (2019) uses adversarial learning methods to disentangle different attributes such as styles of a sentence. Similarly, Radford et al. (2017) trains a LM on a sentiment classification dataset and find that one neuron is responsible for the sentiment value in generation. Our proposed disentanglement methods, on the other hand, encourages the alignment function to encode different attributes to different representations and we leverage Bayes' Rule to further separate attributes.

In multilingual machine translation, a language representation is learned by appending a language code to the source sentence (Johnson et al., 2016) or summing with word embeddings (Conneau and Lample, 2019) to guide the translation towards the target language. Inspired by these methods, `Attribute Alignment` appends the attribute to the beginning of a sentence and learns an attribute alignment function to transform attribute representations while freezing the LM parameters, which is different from previous methods that fine-tune the whole model.

3 Methodology

Unconditional language models are trained to optimize the probability of $p(x_i|x_{0:i-1})$ where x_i is the next token to generate and $x_{0:i-1}$ are already generated tokens. For controlled generation, we need to model the conditional distribution $p(x_i|x_{0:i-1}, \mathbf{a})$ where $\mathbf{a}$ is the attribute for the model to condition on. To make use of large LMs pre-trained on unlabeled data, we need to infuse the attribute $\mathbf{a}$ into the pre-trained unconditional distribution $p(x_i|x_{0:i-1})$. We introduce `Attribute Alignment` for this purpose. Different from fine-tuning the whole LM, our alignment function is the only trainable component in our model while the pre-trained LM parameters are frozen.

3.1 Attribute representation with alignment function (A)

The high-level idea is to append the attribute token to the beginning of a prompt as a signal so that each token in the sentence can attend to the attribute token. However, this may break the originally learned sequential dependencies because now the sentence starts with an attribute token followed by a regular sentence, different from the data used for large LM pre-training.

Instead, `Attribute Alignment` first gets the hidden states of the attribute by running the pre-trained LM on $\mathbf{a}$. Then we align the hidden states using our alignment function ($\mathcal{F}$), implemented as a multi-layer perceptron (MLP) with non-linear connections in this paper, to get aligned attribute representation. Specifically, in the Transformer architecture (Vaswani et al., 2017) where hidden states are represented as key-value pairs, the key (K) and value (V) pair after attribute representation alignment is represented by

$$K'_{:t}, V'_{:t} = [\mathcal{F}(K_{\mathbf{a}}); K_{:t}], [\mathcal{F}(V_{\mathbf{a}}); V_{:t}] \quad (1)$$

$K_{\mathbf{a}}$, $V_{\mathbf{a}}$ are from $LM(x_{\mathbf{a}})$ and $K_{:t}$, $V_{:t}$ are from $LM(x_{:t})$ where $x_{\mathbf{a}}$ is the attribute phrase, and $x_{:t}$ are the tokens in the generated sentence up to timestep t. Then we can calculate attention and output in the original Transformer model.

During training, we freeze the pre-trained LM and compute the language modeling loss on datasets with the attribute $\mathbf{a}$ to train the alignment function $\mathcal{F}$. The loss function is thus

$$\mathcal{L}_A = -\sum_{t=0}^{l} \log p(x_t|\mathbf{a}, x_{:t}) \quad (2)$$

and we only update the parameters of the alignment function using the gradients. Fig.1 illustrates the model architecture. At inference time, all tokens starting from the prompt attend to the target attribute representation transformed by the trained alignment function in addition to the standard self-attention to generate the next token. Intuitively, this can be considered as a conditional LM because all tokens now can attend to the aligned attribute representation.

3.2 Disentangle irrelevant attributes

The learned alignment function bridges the attribute representation to pre-trained LMs. However, we do not disentangle different features in the training data. For instance, if we train the alignment function on a movie review dataset for sentiment control, then $\mathcal{F}$ encodes both sentiment and movie review style after aligning the sentiment attribute representation. Thus, the target attribute representation may be diluted. To solve this problem, we propose three disentanglement methods.

3.2.1 Attribute representation with corpus representation disentanglement (AC)

We propose to add a corpus domain representation $\mathbf{d}$ along with the attribute representation $\mathbf{a}$ during training. For a training corpus (such as movie reviews) with multiple attributes (such as positive and negative sentiment), $\mathbf{d}$ is used in all the training data while $\mathbf{a}$ is only used in a subset of the training data labeled with the target attribute. Similar to Liu and Lapata (2018), this can encourage the model to encode target attribute and other features separately into different representations. Specifically, the key-value pairs can be represented as

$$K''_{:t}, V''_{:t} = [\mathcal{F}(K_{\mathbf{a}}); \mathcal{F}_{\mathbf{d}}(K_{\mathbf{d}}); K_{:t}], [\mathcal{F}(V_{\mathbf{a}}); \mathcal{F}_{\mathbf{d}}(V_{\mathbf{d}}); V_{:t}] \quad (3)$$

where $\mathcal{F}_{\mathbf{d}}$ is a separate alignment function for corpus domain representation, and $K_{\mathbf{d}}$, $V_{\mathbf{d}}$ are from the LM encoding of corpus domain names. Compared to attributes, corpus domain names might be more abstract so we use special tokens for $\mathbf{d}$ (such as <movie review>) and the original texts for attributes (such as `athlete`). At inference time, we ignore the corpus representation while having tokens attend to the attribute representation in addition to normal self-attention as in Equation 1.

3.2.2 KL disentanglement (ACK)

We also experiment with adding KL-Divergence on top of AC to ensure that the LM does not diverge too much from the original distribution when an attribute signal is added. The disadvantage of this method, however, is that KL-Divergence may also prevent the alignment function from learning useful updates to attribute representation.

3.2.3 Bayes disentanglement (ACB)

To further disentangle different features, we use Bayes' Rule to split domain-relevant distribution from attribute-relevant distribution. Derived from Bayes' Theorem (See Appendix A.1), we have

$$p(x|\mathbf{a}) \sim \frac{p(x|\mathbf{a}, \mathbf{d})}{p(x|\mathbf{d})} \cdot \frac{p(x, \mathbf{a})}{p(\mathbf{a}|x, \mathbf{d})} \quad (4)$$

$p(x|\mathbf{a}, \mathbf{d})$ is the probability distribution of the generated sentence conditioning on both the attribute and the corpus domain, while $p(x|\mathbf{d})$ is the probability distribution of the generated sentence conditioning on the corpus domain only. Approximating this equation by eliminating the rest, which can be roughly considered as a constant due to the uniform distribution in the training dataset among different attributes, we can approximate the desired conditional probability in the log space as

$$\log p(x|\mathbf{a}) \sim \log p(x|\mathbf{a}, \mathbf{d}) - \log p(x|\mathbf{d}) \quad (5)$$

During training, we train the attribute and domain alignment functions ($\mathcal{F}, \mathcal{F}_{\mathbf{d}}$) by running the LM conditioned on both attribute and domain

($p(x|\mathbf{a}, \mathbf{d})$), and on domain only ($p(x|\mathbf{a})$). In specific, the loss function is

$$\mathcal{L}_{ACB} = -\sum_{t=0}^{l} \log p(x_t|\mathbf{a}, \mathbf{d}, x_{:t}) - \sum_{t=0}^{l} \log p(x_t|\mathbf{d}, x_{:t}) \quad (6)$$

Similar to other proposed methods, the loss is used to update $\mathcal{F}$ and $\mathcal{F}_{\mathbf{d}}$. At inference time, suggested by Li et al. (2016), we use a hyper-parameter λ to balance the two distributions. Therefore, the distribution we sample tokens from is

$$\log p(x|\mathbf{a}) \sim \log p(x|\mathbf{a}, \mathbf{d}) - \lambda \log p(x|\mathbf{d}) \quad (7)$$

3.3 Zero-shot inference

As we learn the alignment function on the attribute hidden representation from word embeddings instead of learning the attribute representation directly, we can switch in any attribute token at inference time. Therefore, we can choose attributes not seen in the training corpus and generate text conditioned on the new topic as a zero-shot setting.

4 Experiments

We evaluate our proposed methods **A**: using attribute representation only; **AC**: Model A with corpus representation for disentanglement; **ACK**: AC with KL disentanglement; and lastly **ACB**: AC with Bayes disentanglement on sentiment control for thorough comparisons.

4.1 Sentiment Control

Data. We use the Stanford Sentiment Treebank (SST, Socher et al., 2013) as our training data. We choose the sentences with positive and negative sentiment to train our alignment function. We select the same 15 prompts such as "Once upon a time" that were used in prior work, which were originally randomly selected, and are listed in Appendix A.2 (Dathathri et al., 2020).

Baselines. We compare with four baselines. **GPT2** generates unconditioned sentences given the prompts from pre-trained GPT2-medium. The generated sentences are coherent and consistent, but may not capture the target attribute. Its fluency, diversity, and how much the results look like a particular training corpus serve as an upper bound. **GPT2-concat** appends the sentiment token (i.e., `positive`, `negative`) before the prompt. It shares the same motivation as our model (see Section 3.1). **GPT2-finetune** is GPT2 fine-tuned with all the model parameters on the same SST dataset by appending an attribute token to the beginning of a sentence. Its sentiment control score is an upper bound. Similar to ours, the state-of-the-art **PPLM** perturbs pre-trained LMs to incorporate attributes without fine-tuning the LM parameters. It serves as a strong baseline.

4.2 Topic Control

Data. For topic control, we use AG News dataset (Zhang et al., 2015) with four topic attributes ("World", "Sports", "Business", "Sci/Tech") and DBpedia (Zhang et al., 2015) with 14 topic attributes such as "natural place" (see Appendix A.3 for the full list) as our training data. We use the same 20 prompts from Dathathri et al. (2020) (see Appendix A.2). AG News dataset collects news articles whereas DBpedia dataset collects entity definitions from Wikipedia.
Baselines. Note that PPLM uses pre-defined bag of words for topic control so that we cannot use the same dataset for direct comparison. Therefore, we only use the pre-trained LM (**GPT2**) as our baseline while having ablation study among proposed models on sentiment control. We choose the best preforming models from sentiment control for topic control experiments (**AC**, **ACB**).

4.3 Evaluation

We evaluate our proposed methods and baselines on sentiment and topic control generation. We use nucleus sampling (Holtzman et al., 2020) for all the methods at inference time. Please refer to Appendix A.4 for implementation details.

4.3.1 Evaluation Metrics

We evaluate the conditional generation results on fluency, diversity, attribute relevance, and training data corpus resemblance.
Fluency is measured by GPT2-large, a pre-trained external LM, different from the LM we conduct our experiments with (GPT2-medium). We get the average perplexity of the generated sentences (including the prepended prompt). The perplexity score also indicates how much the generated examples diverge from the pre-trained LM.
Diversity is measured by distinct uni-, bi-, and tri-gram ratios as Dist-1, Dist-2, and Dist-3 (Li et al., 2016) averaged over all generated sentences.
Attribute relevance measures how well the generated examples conditioned on the target attributes. For sentiment control, we train an external sentiment classifier using IMDB movie review dataset

(Maas et al., 2011) with a BERT (Devlin et al., 2019) classifier. The classifier achieves an accuracy of 88.51% on the IMDB test set. We also experimented with an internal sentiment classifier trained with SST development set, and we observe that the prediction on the generated texts is similar to that with the external classifier.

For topic control, we train multi-class classifiers with BERT using 80% of the development sets of AG News and DBpedia datasets. The classifiers achieve an accuracy of 89.71% and 99.25% on the rest of the two development sets, respectively. Because other datasets do not share the same topics, we cannot train external classifiers.

The classifiers predict the probability that a given sentence has the target attribute. We report the average attribute-relevant probabilities over all generated examples.

Training data corpus resemblance is used to evaluate if the proposed methods generate sentences that contains undesirable features such as style from the training corpus. For instance, because our proposed method trains with a movie review dataset, the generated examples may tend to be semantically similar to movie reviews. Similar to attribute relevance, we train a BERT classifier by randomly selecting 2,000 training examples and 500 development examples from each of SST, DBpedia, and AG News, and the trained classifier achieves an accuracy of 99.3%. We report the probability that a generated sentence is from its controlling attribute training corpus as the corpus resemblance score.

4.3.2 Evaluation setting

PPLM samples ten sentences and selects the most attribute-relevant one for evaluation to achieve the best performance, and reports the average over three runs. However, we argue that a stable and convincing conditional model should be able to control the generation process for each generated sentence rather than choosing one example from a pool of sentences with high variance. Therefore, we compare the average performance on all the 30 (3×10) conditionally generated results.

With ten times more generated examples for each prompt, results reported by an accurate attribute classifier are reliable. Even though generally preferred, human evaluation may not be ideal here because the generated sentences may associate with the target attribute only due to some nuances such as keywords rather than sentence-level semantics. Compared to subjective humans, attribute relevance classifiers are optimized on the sentence level. In addition, Dathathri et al. (2020) show correlation between automatic classifiers and human evaluations on controlling and language quality. Meanwhile, large scale A/B test with multiple models may not be feasible. Therefore, to make the results more repeatable, we choose to use automatic evaluation metrics.

5 Results and Analysis

We show controlled examples in Table 1 and analyze sentiment and topic control results as follows.

5.1 Sentiment control

Comparison with baselines. Table 2 shows results on sentiment control. Compared to the pre-trained LM (GPT2, 49.24%), all our proposed methods achieve better sentiment controlling scores with a large margin and get similar distinct scores. This shows that our proposed method is effective in sentiment control.

Even though GPT2-finetune achieves the highest sentiment score (78.78%), it gets higher perplexity, lower distinct scores, and very high corpus resemblance (92.24%). This implies that we can fine-tune a pre-trained LM to condition on the target attribute but suffer from the cost of being restricted to generating sentences resembling the training data as motivated by Section 1.

All our disentangled methods outperform PPLM with better sentiment control and diversity while having lower perplexity. We note that PPLM uses top-k sampling and the sampling method may result in different performance. To eliminate the influence from sampling methods, we also compare our methods with PPLM by top-k sampling and our methods show higher sentiment probability and lower perplexity with the same trend (see Table 4 in Appendix A.5). For qualitative comparisons between our proposed method and PPLM, we use the IMDB classifier to rank the most negative sentence generated from 30 examples for each prompt and show the generated results in Appendix A.7. Compared to our models, PPLM suffers from repetition and degeneration problems suggested by both distinct scores and qualitative analysis from the generated examples. Results show that regardless of the sampling method, using the aligned attribute representation as a control signal to guide the text generation leads to higher sentiment con-

Model	Attribute			Quality				Data
	Sentiment% ↑	Positive% ↑	Negative% ↑	PPL↓	Dist-1 ↑	Dist-2 ↑	Dist-3 ↑	Corpus resemblance % ↓
Baselines								
GPT2	49.24	77.15	21.33	37.78	**0.49**	**0.85**	**0.91**	**18.31**
GPT2-concat	52.24	65.42	39.06	57.5	0.49	0.84	0.89	18.87
PPLM	57.03	81.58	32.47	54.03	0.44	0.79	0.88	26.12
Attribute Alignment								
A	52.61	77.35	27.86	40.19	0.45	0.82	0.90	59.13
AC	68.92	80.22	57.61	48.78	0.47	0.84	**0.91**	62.13
ACK	64.89	76.25	53.53	52.66	0.48	0.84	**0.91**	62.8
ACB	64.49	85.35	43.64	36.62	0.48	**0.85**	**0.91**	24.05
Attribute Alignment with strong polarized training data								
AC-S	67.04	81.62	54.45	38.46	0.45	0.80	0.88	63.21
ACB-S	58.85	80.88	36.82	**33.33**	0.46	0.83	0.89	28.12
Language model fine-tuning								
GPT2-finetune	**78.78**	81.92	75.63	55.60	0.37	0.66	0.75	92.24

Table 2: Results on sentiment control. Sentiment probability indicates how likely the generated sentence has the target sentiment attribute predicted by an external classifier averaged by all generated examples. Corpus resemblance indicates how much the generated results read like movie reviews. Our proposed model with Bayes disentanglement (ACB) achieves good performance on sentiment controlling while maintaining high quality language generation. Note that even though GPT2-finetune achieves the best sentiment controlling score by training the whole LM, it suffers in generation quality and the generated sentences read like movie reviews measured by corpus resemblance.

trolling probabilities while keeping the original LM distribution compared to PPLM.

Comparison among proposed methods. The worse performance of having attribute representation only (52.61%) indicates that the entangled attributes dilute the conditional distribution and result in texts using similar vocabularies suggested by low diversity scores. In comparison, adding a corpus representation to disentangle target attributes leads to the best performance on sentiment probability prediction. Further disentanglement by adding KL-Divergence and separating corpus distribution with Bayes' theorem helps to reach lower perplexity and higher distinct scores as expected, but it hurts the attribute controlling performances. This may be caused by that the attribute and corpus representations in fact still mingle with each other so that when we remove the corpus distribution, we also remove some of the target attribute distribution. We also note that without Bayes disentanglement, all the other proposed methods reach much higher training corpus resemblance score (e.g. 62.13% with AC) but still much lower than that from fine-tuning (92.24%). This may be partially explained by that sentences with a strong sentiment are more similar to movie reviews than the neutral DBpedia and news data from the training corpus resemblance classifier. Combining all the metrics, it shows that there is trade-off between sentiment control and generation quality. However, we can still control the sentiment better without the cost of perplexity, diversity, and style convergence than the strong baselines.

Attribute data influence results. To evaluate how much attribute relevance in training data influences controlling effect, we experiment with training on strong polarized examples labeled as "very positive" and "very negative" from SST. We denote the corresponding models as **AC-S**: AC with strong polarized training data; and **ACB-S**: AC-S with Bayes disentanglement. Table 2 shows that training with strong polarized data achieves similar controlling ability but suffers from lower diversity. This suggests that our proposed method is not sensitive to the attribute quality in the training data, showing the potential to use less strictly annotated data for controlling more diverse attributes.

Adversarial prompts results. Following Dathathri et al. (2020), we also experiment with generating a sentence to an opposing sentiment from a highly polarized prompt. For example, the goal is to generate a positive sentence with the negative prompt "The food is awful" and a negative sentence with the positive prompt "The food is amazing", respectively. Using the external classifier to select the generated examples with the

Topic source	Model	Attribute	Quality				Data
		On topic prob. % ↑	Perplexity↓	Dist-1 ↑	Dist-2 ↑	Dist-3 ↑	Corpus resemblance % ↓
AG News	GPT2	25.43	38.00	**0.49**	**0.84**	**0.90**	**84.08**
	AC	63.38	32.37	0.47	0.83	**0.90**	92.47
	ACB	**64.80**	**31.22**	0.46	0.83	**0.90**	90.66
DBpedia	GPT2	6.63	**37.40**	0.49	**0.84**	**0.90**	**1.01**
	AC	**32.98**	60.22	**0.50**	**0.84**	**0.90**	7.33
	ACB	32.18	49.85	0.49	0.83	**0.90**	7.46

Table 3: Topic control results with topics from AG News and DBpedia. On topic probability indicates the probability that the generated sentence is conditioned on the target topic attribute predicted by a topic classifier averaged by all the examples. Our proposed methods outperform the baselines by a large margin while having similar perplexity and diversity compared to the pre-trained language model (GPT2).

most likely target sentiment, we obtain:
positive: The food is awful but the service is amazing! The takeout is amazing! However, for me, a small, cozy restaurant that is a small institution in a small town I'm so glad that they are planning on ...
negative: The food is amazing!! We didn't want to bring it home as the night before, but we can't afford the honey pot cost so we ended up throwing in our own trail mix as well as having ...

Despite the prompts being very polarized, our method can still lead the text generation to the target sentiment without compromising fluency and diversity. More importantly, although we train our alignment function in the movie review domain, our generated sentences are not biased towards the domain. We show comparisons to PPLM in Table 5 in the appendix.

5.2 Topic control

Comparison among different methods. We present our results on topic control in Table 3. Similar to sentiment control, we observe that our proposed methods significantly outperform the baseline in target topic controlling while holding similar perplexity and distinct scores. In addition, using Bayes' disentanglement results in lower perplexity. However, compared to the sentiment control, further disentanglement derives controlling effect on par with the simple disentanglement ($+1.42\%$ and -0.80% relative change for AG News and DBpedia) and generates comparable distinct scores. The finding is similar to corpus resemblance. This indicates that topic attribute representations may be less entangled with other features such as style from the training corpus compared to that for sentiment representation.

Comparison between training dataset. To compare the results between topics from AP News and DBpedia, the perplexity is higher than the baseline and the relative corpus resemblance score is also high for DBpedia. We conjecture that this is caused by that topics such as "educational institution" may be difficult to associate with prompts such as "Emphasised are" in the pre-trained LM. When we control the model to generate sentences with the corresponding attributes, the generation diverges from the pre-trained LM more. However, distinct n-grams are not sacrificed.

Zero-shot analysis. Topics such as "military" are not in the topic control training corpus so that they are considered as zero-shot attributes. We show examples in Table 1. Our trained alignment function can map unseen attribute representation to the target representation to generate fluent and on-topic sentences. However, this zero-shot ability largely depends on the unseen attribute and the provided prompt. We leave careful evaluation on aligning unseen attribute representation to future work.

6 Conclusion

In this paper, we propose a simple but effective attribute alignment model for conditional language generation on top of non-controlled pre-trained LM without fine-tuning LM parameters. We also introduce disentanglement methods to separate different features from the training corpus to further preserve the original pre-trained LM distribution. Evaluated on sentiment and topic control, we show that our proposed method outperforms the previous methods on attribute control while maintaining language generation quality. For future work, we plan to apply the proposed methods on other attributes such as dialog act and explore few-shot learning settings of the training corpus.

7 Ethical Considerations

The proposed method is intended to explore approaches to perturb pre-trained large language models. We hope that our method can inspire future research on conditional generation while maintaining the original LM generation quality. Meanwhile, we note that our method can be used to generate negative sentences which may harm some use cases. However, similar to previous research, we can apply our method to control the generation to less toxic directions and reduce the risks of misuse. In addition, our experiments are done on English data, but our method can be applied to any language. We did experiments with the same setting and same data with previous research when we claim better performance.

References

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

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069. 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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

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.

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

Cong Duy Vu Hoang, Trevor Cohn, and Gholamreza Haffari. 2016. Incorporating side information into recurrent neural network language models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1255, San Diego, California. Association for Computational Linguistics.

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*.

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

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of Machine Learning Research*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, California, USA. PMLR.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

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

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.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

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.

Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444.

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

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

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

Alexey Romanov, Anna Rumshisky, Anna Rogers, and David Donahue. 2019. Adversarial decomposition of text representation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 815–825, Minneapolis, Minnesota. 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, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational auto-encoder for text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 166–177, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

A Appendices

A.1 Bayes Theorem Proof

By Bayes' Theorem,

$$
\begin{aligned}
p(x|a,d) &= \frac{p(x,a,d)}{p(a,d)} && (8)\\
&= \frac{p(d|x,a)\cdot p(x,a)}{p(a,d)} && (9)\\
&= \frac{p(d|x,a)\cdot p(x|a)\cdot p(a)}{p(a,d)} && (10)\\
&= \frac{p(d|x,a)\cdot p(x|a)\cdot p(a)}{p(d|a)\cdot p(a)} && (11)\\
&= \frac{p(d|x,a)\cdot p(x|a)}{p(d|a)} && (12)\\
& && (13)
\end{aligned}
$$

so that we can get

$$
p(x|a) = \frac{p(x|a,d)\cdot p(d|a)}{p(d|x,a)} \quad (14)
$$

Transforming the denominator by

$$
\begin{aligned}
p(d|x,a) &= \frac{p(x|d)\cdot p(d)\cdot p(a|x,d)}{p(x,a)} && (15)\\
&\propto \frac{p(x|d)\cdot p(a|x,d)}{p(x,a)} && (16)\\
& && (17)
\end{aligned}
$$

we can get

$$
\begin{aligned}
p(x|a) &\sim \frac{p(x|a,d)}{p(x|d)}\cdot\frac{p(x,a)}{p(a|x,d)}\cdot\frac{p(d|a)}{p(d|x,a)}\\
&\sim \frac{p(x|a,d)}{p(x|d)}\cdot\frac{p(x,a)}{p(a|x,d)}
\end{aligned} \quad (18)
$$

A.2 Prompts for Experiment

We use the same 15 prompts used for sentiment control experiment and 20 prompts used for topic controlling experiment from PPLM (Dathathri et al., 2020).

Sentiment control: “Once upon a time", “The book", “The chicken", “The city", “The country", “The horse", “The lake", “The last time", “The movie", “The painting", “The pizza", “The potato", “The president of the country", “The road", and “The year is 1910.".

Topic control: “In summary", “This essay discusses", “Views on", “The connection", “Foundational to this is", “To review,", “In brief,", “An illustration of", “Furthermore,", “The central theme", “To conclude,", “The key aspect", “Prior to this", “Emphasised are", “To summarise", “The relationship", “More importantly,", “It has been shown", “The issue focused on", “In this essay".

A.3 DBpedia topics

The 14 topics from the DBpedia dataset are: "company", "educational institution", "artist", "athlete", "officeholder", "means of transportation", "building", "natural place", "village", "animal", "plant", "album", "film", "written work". (Zhang et al., 2015)

A.4 Implementation Details

We use GPT2-medium (Radford et al., 2019) with 355M parameters as our pre-trained language model, and GPT2-large with 774M parameters as an external language model to evaluate perplexity. Our implementation is based on an efficient transformer architecture (Wolf et al., 2019) where hidden states are stored as key-value pairs. We implement the alignment function with a multi-layer perceptron (MLP) of two linear layers and a nonlinear activation function (ReLU). Both at training and inference time, all tokens in the sentence can attend to the attribute representations as if they are appended to the beginning of the sentence, but we fix the position ids of the sentence to start with 0. We apply nucleus sampling (Holtzman et al., 2020) with p set to 0.9 and generate texts with a maximum length of 40 for all the experiments.

We did not do exhaustive hyperparameter search. For λ used in **ACB**, we tried $0.1, 0.5, 1$. We choose the best hyperparameters on a held-out set of prompts using the evaluate metrics. We set $\lambda = 0.1$ and report the results in the paper. Similarly, we experimented with $0.01, 0.1, 1$ for the KL scale and show results in the paper with KL scale set to 0.01. However, we use the suggested hyperparemters from the paper and code for the baselines we compare with. On SST training set for sentiment control, each epoch takes about 250 seconds, 720 seconds, 720 seconds, and 720 seconds for **A**, **AC**, **ACK**, and **ACB** respectively on a RTX 2080 Ti GPU machine. We train for 50 iterations for each model. It takes about 3.5 seconds to generate 30 examples for each prompt with evaluation on proposed evaluation metrics.

A.5 Comparison using top-k sampling

Model	Sent. prob.% ↑	Perplexity↓
GPT2	49.98	**10.94**
PPLM	58.57	17.52
AC	**67.39**	16.53
ACB	60.54	13.35

Table 4: Comparison on different methods using top-k sampling ($k = 10$).

A.6 Comparison to PPLM on adversarial prompts

Model	Target	Generated Text
ACB	positive	The food is awful but the service is amazing! The takeout is amazing! However, for me, a small, cozy restaurant that is a small institution in a small town I'm so glad that they are planning on ...
	negative	The food is amazing!! We didn't want to bring it home as the night before, but we can't afford the honey pot cost so we ended up throwing in our own trail mix as well as having ...
PPLM	positive	The food is awful but there is also the music, the story and the magic! \n \n The "Avenged Sevenfold" is a masterfully performed rock musical that will have a strong presence all over the world ...
	negative	The food is amazing\n \n It's also not. \n \n It is not the kind of stuff that you would just want to spend your money on for \n \n I don't know why, but when I got my second box, it felt like a bad rip off ...

Table 5: Adversarial sentiment control examples compared to PPLM generated from the proposed alignment function with Bayes disentanglement (ACB) where the prompt has a strong opposite sentiment. Similar to Table 1, the results here are selected by a sentiment classifier (not cherry-picked). PPLM results are taken from PPLM paper (Dathathri et al., 2020)

A.7 Comparison between Attribute Alignment (ACB) and PPLM Examples

Model	pred%	PPL	Generated Text
PPLM*	98.31	22.19	Once upon a time \n\n I made this game for my wife, and she loved it! I have made a wonderful discovery of how to make this very amazing and beautiful looking and beautiful, beautiful, amazing book! I
PPLM	98.39	119.54	Once upon a time, in a distant galaxy, a supernova blast destroyed a supernova explosion the losing side ripping apart sScRush UV-3a. A burnt out and rusty mess of garbage spoods the
ACB	99.52	42.17	Once upon a time, eBay lists its canceled items. I don’t think there is a list of canceled items that I can see here. In the meantime, a bunch of crap, from iPhones (minus their selling center
PPLM*	96.53	13.52	The city of Detroit, the country’s third-richest and most-populous, is the most violent, most dysfunctional and most pathetic city in American history; that is, if the United States, which
PPLM	99.88	158.57	The city might as well have been written by \n\n "We got into this mess, how could youWhat. and by" (by the night was "O-but of the/-how we"
ACB	98.07	31.98	The city is a city of commerce, traffic and construction. In the midst of all this noise and bustle, West Queen West has produced the most monotonous of urban mini-revolutions: no
PPLM*	99.46	11.67	The book was not the best. I found some of the writing to be slightly clunky and awkward. I liked some, but not enough to buy a copy. \n\n The plot seemed to be about a
PPLM*	99.84	29.16	The book comes out of the ashes of my writing and writing. It was a complete disaster from beginning to end. I had seasoned 250,000 words–at least an hour’s per day to write on it
ACB	99.69	27.30	The book was poorly written, written off as ’opinion’ and poor grammar and punctuation were used extensively. No wonder the author is currently living in a halfway house with his estate. Nevermind the fact
PPLM*	83.42	21.89	The president of the country’s largest hospital says they are now on a "mission to save lives" and that "the people of the US" should not pay for this. I’m a man. I am the reason the
PPLM	86.46	32.13	The president of the country’s largest college was fired from her job for giggling at a Golden Gate crowd as a crowd in San Francisco rallied to help a storm victim. \n\n However, the New York Times shames the
ACB	97.93	13.32	The president of the country isn’t in office. The president isn’t in office. You can’t make an argument based on that kind of stuff. So what’s the point of it?
PPLM*	69.73	21.67	The painting will be on the back \n\n - "A \n\n . \n\n - n\n\"< |endoftext| >"I can’t be the only one who has the right to be a ’f*ck you
PPLM	99.44	119.54	The painting is thought fluff, a very poor, and a shambolic, modern, and bannister-ly, why did you just do that to me, you and your wretched brand of dreadful
ACB	99.75	20.76	The painting is one of the worst I’ve seen in my lifetime... it’s so corny and flat. It’s such a cheap, offbeat example. It’s more shocking than shocking, because you wouldn

Model	pred%	PPL	Generated Text
PPLM*	96.92	73.73	The horse has no need for any of this. \n\n ; ; ; : : ; ; ;; ; ;! # #!? :? *? :? no (: the ((@ the (
PPLM	97.72	94.93	The horse is a wyvern. A wyder is a "rifle". A good shot. Create Chris C, a pretty, brunette, a skinny, bald drone. Just a fat.
ACB	97.39	47.43	The horse he’s teaching to lick it away at the bar: heck, the economy would be better off if they didn’t have one. In fairness, he could certainly have cut some of his cast more slack
PPLM*	94.61	20.66	The lake has long been the center for a long, ugly, and and and and. \n\n . \n\n . \n\n . \n\n . \n\n The problem with the problem is I can’t find
PPLM	96.21	43.00	The lake around Yaffo in south-central Russia in the world’s only biodiversity-poor desert was the scene of the worst air quality in Europe, with more than half of the population suffering three different types
ACB	97.39	44.15	The lake is not vast enough to accommodate a tight lake liner. \n\n 1. Looking for catnap materials in the lake \n\n Finding a catnap bather or two is like asking a family
PPLM*	98.72	15.78	The country is in a tailspin with the economy barely growing and the budget deficit rising. \n\n The government’s budget is a failure. A failure for which there’s nothing the public will not pay the price
PPLM	97.20	32.84	The country’s will merely sit silently on its grave. \n\n A federal government miscalculated and the economy is limping back to the roots. \n\n Just how bad are the latest developments and what do
ACB	94.53	38.63	The country has become too interested in its politics to pay attention to anything else. The top domestic TV stations should say nothing about this conflict or this nation and instead should be focusing on discussing the place of gays and
PPLM*	97.83	84.65	The road to the White House is an ugly,,,,,,,,,,... \n \n A house or a bin is a a a. It’s got an awful, unpleasant name.
PPLM	86.51	33.87	The road to intensive genetic counseling \n\n When can an individual apply for a program under "animal welfare" or "conservation"? \n\n What does "social welfare," "mangle" and "population
ACB	97.58	35.38	The road trip to Seattle and Pacific Ocean was plagued with weird mishaps. Our airline plane couldn’t land at Seattle airport so a Toyota Hilux truck was hired to drive it, and a lot of the freight
PPLM*	99.86	8.94	The movie is a total disaster, and it’s been a total failure at that. \n \n I can’t say I’m entirely convinced it’s been a total failure, but I can assure you I’m not
PPLM*	99.77	34.70	The movie is chock-full of nothing good. It is a horrible 3 minute movie. It would’ve been terrible if I didn’t see it when I was through Foety.
ACB	99.60	23.77	The movie set/store just makes no sense at all. \n \n I’ve only had my phone with me for a week, so I didn’t watch it while others did. On day two I caught the
PPLM*	97.65	25.52	The pizzaiolo or specialised freezer version of an Italian classic is no better or worse than a standard hot dog. The good, you just eat it, while the bad, you cook it way, way
PPLM	99.64	20.79	The pizza oven. The pan. What a boring, boring job. You put everything in it, right? So there’s this wonderful smell in there. But this is the worst part of it. It takes
ACB	99.18	25.45	The pizza box is, in a word, a piece of garbage \n The first-ever-to-make-it-in-a-Bowl \n An over-sucking, over-dram

Model	pred%	PPL	Generated Text
PPLM*	95.21	8.33	The last time New Orleans faced a top-five opponent, it went 4-13. \n \n The Pelicans have lost six consecutive games. \n \n The Pelicans have the league's fourth-worst offense.
PPLM	47.17	27.44	The last time I wrote about buryable garbage I compared the North American garbage garbage heap to that of "Hell" on Earth, and if you have been following along you have noticed some large differences. "Hell"
ACB	97.10	15.95	The last time we were all excited about Wolf of Wall Street, it was Sam Jackson. It took for us a while for us to realize that this was going to be a terrible idea. \n \n Stereotype
PPLM*	99.15	22.28	The year is 1910. A woman, the only child in a small, isolated village, is attacked and is killed.< |endoftext| >I have no time for the idiotic and disgusting bullshit that is the NFL, and that is just
PPLM	96.24	39.15	The year is 1910. Colonists on a long-sought-after research mission return to a barren world of dirt and rubble. The expedition discovers a barren, randy device who possesses a hundred-year-old device
ACB	99.93	17.29	The year is 1910. He's going back home to Paris, where he's an English salesman. He's trying to raise a family and he's having some trouble when his wife returns from an extended vacation.
PPLM*	99.30	14.63	The potatois the world's most widely eaten meat, and its high price is why we eat so much. But is the potato actually the worst meat you'll get? And does a potato really have the worst
PPLM	99.00	166.03	The potato, a slender, poorly vascular plant that is a poor choice for many traditional timesaving reasons. Full of nasty things like the inability to remember details where the it is raised, is the sention the
ACB	99.52	66.44	The potato seems to be a slow, vomiting, and hungry thing. I have seen it eat its excess of juice and poop and drink in thin streams. Yet, despite this hideous abnormality, it hardly feels
PPLM*	98.47	18.20	The chicken wing virus was a terrible thing. I mean, really bad. \n \n The virus, known as "Chicken Wing," was a disease that was devastating to the entire chicken world, killing thousands of chickens
PPLM	95.96	26.20	The chicken coop is a great idea for people, but if you are getting pregnant, the plan is not going to work. Hermies, baby and toddlers are at risk. \n\n Most people would
ACB	99.24	39.75	The chicken commercial is packed full of even more bullshit. For the nearly 900th time, Wendy's CEO Joe Noller has made it clear that there is an organization in this country that hates its products, specifically

Table 6: Examples from PPLM(Dathathri et al., 2020) and our proposed method (ACB: attribute and corpus representation with Bayes disentanglement) for each prompt we experiment with. Note that the perplexity is not comparable among different sampling methods. We use top-p sampling for ACB and and PPLM, and top-k sampling for PPLM* because Dathathri et al. (2020) suggests top-k in their paper for the best results. We use an external classifier to select the example with the highest negative probability from 30 generated sentences and present the results.