Prompt Compression and Contrastive Conditioning for Controllability and Toxicity Reduction in Language Models

David Wingate
Brigham Young University*
wingated@cs.byu.edu

Mohammad Shoeybi
Nvidia, Inc.
mshoeybi@nvidia.com

Taylor Sorensen
University of Washington†
tsor13@cs.washington.edu

Abstract

We explore the idea of compressing the prompts used to condition language models, and show that compressed prompts can retain a substantive amount of information about the original prompt. For severely compressed prompts, while fine-grained information is lost, abstract information and general sentiments can be retained with surprisingly few parameters, which can be useful in the context of decode-time algorithms for controllability and toxicity reduction. We explore contrastive conditioning to steer language model generation towards desirable text and away from undesirable text, and find that some complex prompts can be effectively compressed into a single token to guide generation. We also show that compressed prompts are largely compositional, and can be constructed such that they can be used to control independent aspects of generated text.

1 Introduction

Language models (LMs), such as GPT-2 (Radford et al., 2018, 2019a), BERT (Devlin et al., 2018), T5 (Raffel et al., 2020), or GPT-3 (Brown et al., 2020), exhibit a remarkable ability to capture patterns of grammar, vocabulary, cultural knowledge, and conversational rhythms present in natural language. Formally, a LM is a conditional distribution over tokens $p(x_t|x_1, \cdots, x_{t-1})$, with each token $x_t \in \mathcal{V}$ for some vocabulary $\mathcal{V}$. Throughout this paper, we will refer to $x_h = x_1, \cdots, x_{t-1}$ as the *prompt*.

This paper explores *prompt compression*: the idea that the text x_h used to condition a LM can be approximately represented by a much smaller set of carefully chosen weights, using the framework of soft prompts (Lester et al., 2021). We begin by establishing some basic properties of compressed prompts, and importantly show that while highly compressed prompts lose fine-grained information about the prompt, they can retain general, abstract information. This motivates our central application: to use such compressed prompts in a Bayesian attribute framework to steer text generation, with specific application to toxicity reduction.

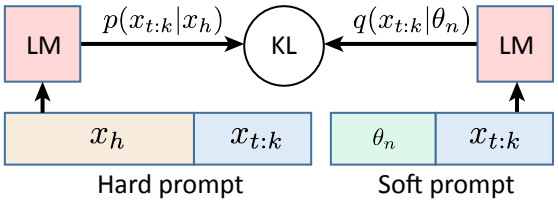

Figure 1: Schematic of prompt compression. Weights of the soft prompt are tuned to minimize the KL divergence between hard and soft prompts, for all $x_{t:k}$.

To motivate this more deeply, we briefly sketch how compressed prompts can be used in toxicity reduction. Efforts to reduce toxicity and bias generally follow one of two strategies: the first is to train or fine-tune LMs on carefully curated data, either tagging or labelling it in special ways (Keskar et al., 2019a; Lu et al., 2022) or using data known to be "clean". The second is to "steer" the generation of token probabilities away from toxic generations (Krause et al., 2020; Liu et al., 2021), and towards text with known, desirable properties.

Following previous work, we steer LM probabilities by using a Bayesian attribute classifier framework that involves scoring candidate tokens with different experts. As an independent contribution, we explore the idea of simply using conditioning text to construct such experts by leveraging the few-shot modeling abilities of LMs (Radford et al., 2019a; Brown et al., 2020): given a few examples of text containing a pattern of interest, language models are capable of "analyzing" such examples and assign high probability to subsequent text exhibiting the same pattern. Thus, in the same way that language model can, for example, clas-

* Work done while at Nvidia, Inc.
† Work done while at Brigham Young University

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5650 - 5663

sify the sentiment of a tweet, we use LMs to analyze the toxicity of candidate generations in real-time. Our method can be considered an exemplar-based method of defining experts that capture desirable and undesirable attributes of generated text. We term this technique *contrastive contexts*, and note that it reduces the problem of creating experts to one of prompt engineering (Reynolds and McDonell, 2021).

However, our conditioning contexts are quite large, which motivated this work. We use prompt compression to mimic an uncompressed prompt (hereafter referred to as "hard" prompt) as closely as possible, thereby saving both computation and space in the context window. Our results demonstrate that this can be very effective, and, in a very surprising finding, that complex prompts can be reduced to a single token and still be useful for toxicity reduction, often with better fluency compared to hard prompts.

The contributions of this paper are three-fold: first, we introduce and formalize the idea of prompt compression; second, we introduce and formalize the method of contrastive contexts in the Bayesian attribute framework; third, we experimentally evaluate our methods, and refine the technique based on various empirical observations, and contribute a careful study of effectiveness as model size varies. Our code is available online. [1]

2 Background and Related Work

To the best of our knowledge, this is the first work to directly explore prompt compression. However, our work is based on the original soft prompt ideas of (Lester et al., 2021). It is also somewhat related to distillation, where one model is trained to mimic another by matching logits (Gou et al., 2021).

Usually, LMs take text as input, which is then tokenized into discrete tokens by a tokenizer. Each token is then mapped to a learned embedding, which is used as input to a transformer (Vaswani et al., 2017). The idea of soft prompts (Lester et al., 2021) is to bypass the need to use discrete tokens with pre-trained embeddings and instead directly learn a series of embeddings via backpropagation. These learned embeddings are then fed directly to the transformer and do not need to correspond to any actual language tokens.

As the centerpiece application of prompt compression, we explore generative controllability (Keskar et al., 2019b) and toxicity reduction in language models.

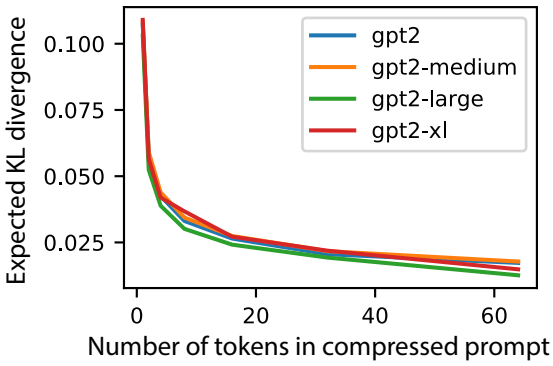

Figure 2: KL divergence of compressed prompts as a function of number of tokens n. Prompts are randomly sampled from the Pile (mean words= 916, median words = 274, median characters = 1849).

Our method is most closely related to decode-time algorithms, such as GEDI (Krause et al., 2020), which uses Bayes' rule and discriminative models to steer the generation towards a certain attribute; and PPLM (Dathathri et al., 2019), which uses an estimated gradient with respect to the desired attribute to steer the LM's internal representation at generation time.

Other methods are based on fine-tuning language models with the classical language modeling objective to steer generation. DEXPERTs (Liu et al., 2021) combines experts and anti-experts in a product of experts model to reduce toxicity of LMs.

Additionally, reinforcement learning approaches show strong performance at steering language models (Stiennon et al., 2020). By providing rewards, methods such as PPO (Schulman et al., 2017) and Quark (Lu et al., 2022) represent the current best performance at reducing LM toxicity while maintaining fluency. These methods, however, require a predetermined reward function, which may or may not be feasible depending on the context.

3 Prompt Compression

Here, we introduce and explore the idea of *prompt compression*, whereby the parameters of a soft prompt (Lester et al., 2021) are trained to mimic a fixed hard prompt as closely as possible.

The intuition of our idea is simple: conditioning a LM on a hard prompt x_h induces a distribution $p(x_t, \cdots, x_{t+k}|x_h)$ over all possible sub-

[1] https://github.com/BYU-PCCL/prompt-compression-contrastive-coding

sequent sequences of tokens $x_t, \cdots, x_{t+k}$ for all k. To simplify notation, let $x_{t:k} = x_t, \cdots, x_{t+k}$. The schematic of the idea is shown in Fig. 1. Formally, a soft prompt is a block of weights θ_n that is prepended to the embeddings of the tokenized sequence $x_{t:k}$, and which is then fed through the transformer layers of the language model. The soft prompt induces a modified distribution over $x_{t:k}$, which we represent as $q(x_{t:k}|\theta_n)$. Here, n is the number of tokens in the soft prompt (which do not necessarily correspond to natural language tokens).

To compress prompt x_h, we train the soft prompt weights to minimize the following objective:

$$\min_{\theta_n} \mathbb{E}_{x_{t:k}} \left[\mathrm{KL}(p(x_{t:k}|x_h)||q(x_{t:k}|\theta_n))\right] \quad (1)$$

where the sequences $x_{t:k}$'s are sentences of various lengths and content drawn from a diverse training set. We optimize this objective using the Adam optimizer for 75,000 steps of training with a linear learning rate schedule starting at 0.1, and $x_{t:k}$'s drawn randomly from The Pile (Gao et al., 2021), requiring about 1-4 GPU-hours to train a single prompt, depending on computational complexity of running the LM. All prompt training was done using either a single A100 or V100 GPU.

While training a compressed prompt can be expensive, the gains are found at inference time. Using a compressed prompt over a hard prompt reduces the length of the context. This scales down the needed computation according to the transformer's attention mechanism, which is $O(n^2)$. This also could allow long contexts to be compressed and appended to longer inputs than was previously possible. Once trained, these compressed prompts could be shared to create a library of efficient contexts.

4 Experiment Set #1: Establishing Basic Properties of Compressed Prompts

We begin by exploring the properties of compressed prompts. First, we show that conditioning on a hard prompt and its compressed prompt generate qualitatively similar generations, although this equivalence degrades as the prompt is compressed more and more. Second, we qualitatively explore what happens to fine-grained information as a prompt is compressed more and more.

Models and codebase. All experiments were conducted using the Huggingface[2] (Wolf et al., 2019) implementation of GPT-2 (117M parameters), GPT-2 medium (345M), GPT-2 large (774M) and GPT-2 xl (1.5B) models.

[2]https://github.com/huggingface/transformers

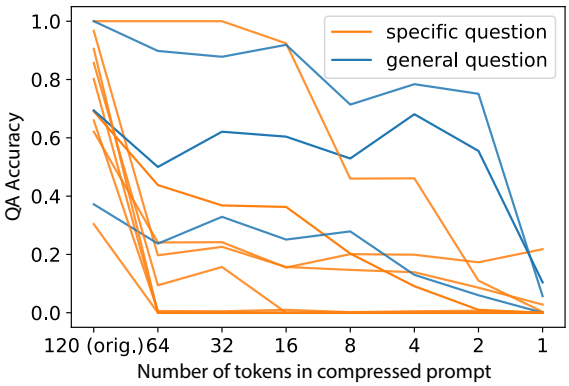

Figure 3: Reading comprehension performance by question as context is more and more compressed. Accuracy is averaged over 1000 completions and each line represents a single question. As expected, performance degrades nearly monotonically as the number of tokens in the compressed prompt is decreased. General questions degrade less than questions about specific details. We used GPT-2 xl for this experiment.

4.1 Comparing hard and compressed prompts

Fig. 2 shows the KL divergence between the original prompt and the compressed prompts' output distribution for randomly sampled sentences from the pile (Gao et al., 2021). As the figure shows, as the size of the compressed prompt increases, the KL divergence monotonically decreases for all models. This implies, as expected, that the more context allowed in a soft prompt, the better the soft prompt does at mimicking the full context.

Additionally, note that the magnitude of the KL divergence is similar across models for a given soft prompt size n. This shows that this method of context compression works well on a variety of model sizes (124M - 1.5B parameters).

4.2 Exploring information retention

As a prompt is compressed more and more, information in the original prompt must be lost. As the training process specifically attempts to match the predictive distribution over completions for a prompt, the question arises: what information is preserved, and what is discarded?

Reading Comprehension Task. To assess this, we construct the following experiment: given a reading comprehension task that involves a paragraph p of text and a series of questions, how do the answers to those questions degrade as a function

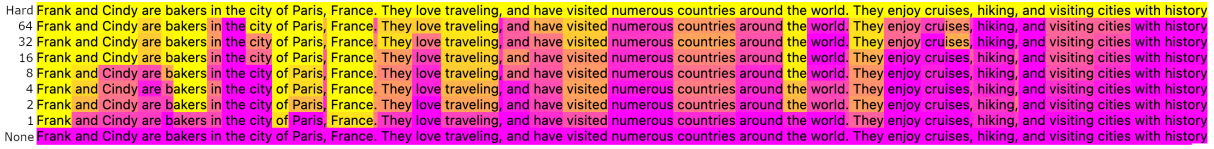

Figure 4: Assessing the information retained as a prompt is compressed more and more severely. The model is tasked with recovering the passage given a hard prompt (the passage), compressed prompts, or no prompt. For each token, likelihood is calculated and scaled so that the probability according to the hard context is 1 and the probability with no context is 0. It is visualized with a heatmap, where yellow corresponds to 1 (hard context) and pink corresponds to 0 (no context).

of compression? Specifically, we look at questions about *fine-grained* information (specific details that occur once) in p, as well questions about *general* information (common themes of the passage that occur multiple times) in p. For the paragraph p and questions used, see Appendix B.

In Fig. 3, we see that prompt compression attempts to retain the more general information about a prompt, even while it cannot retain fine-grained details. Additionally, as we would expect, the more compression happens, the more information from p is lost. In the next section, we will see that this property will be useful in the context of toxicity reduction.

Reconstruction Task. Another way to measure the information retained is to task the language model with reconstructing the paragraph. Specifically, given a soft context of a certain length, we append the prompt "Now repeat the text:". We then look at the likelihood of each token in the text, normalized between the baselines of no compression ("hard" context) and no context at all (note that some words may be easily predicted simply by virtue of grammatical rules of English). Results are shown in Fig. 4. For the heatmap over the full paragraph, see Appendix D.

As expected, as n decreases, so does the amount of retained information about the paragraph p. The largest soft prompt ($n = 64$) seems to retain information primarily about the following tokens: "Frank and Cindy", "bakers", "city of Paris, France", "They love travelling", "visited", "countries," "cruises," etc. At the lowest size of $n = 1$, most of the information is lost, but the model still predicts "Frank", "of", and "France" with significantly higher probability than having no context. Qualitatively, it also appears that, at least for this prompt, information earlier in the prompt is retained better than information later in the prompt.

4.3 Compositionality of compressed prompts

Here, we briefly explore the idea that multiple severely compressed prompts can be combined to modulate different properties of generated text.

To do this, we use two contexts: one that primes the LM for negative sentiment, and a second that primes the model for talking about cats (both contexts can be found in Appendix A). We then test the effect of steering the model towards different types of text by conditioning on a context which is either negative, talks about cats, or both.

In this experiment, we prompt the model with "I thought the movie was," trying to prompt the model to output a movie review. As you can see in Table 1, when you use none of the contexts for conditioning, the baseline level of prompts generated that are about cats or with negative sentiment is low. As you condition on the (normal or soft) contexts individually, the number of completions which contain negativity and cats respectively increase. This shows the efficacy of the in context style transfer for both attributes, using either soft or "hard" prompts. Finally, when you concatenate the two contexts, you see behavior somewhere in between the baseline and the individual completions. This shows that to some degree, you can compose these soft prompts together to steer completion behavior.

Interestingly, the prompts that best elicit negative sentiment and sentences about cats are the compressed prompt versions. This suggests that the compressed prompt may capture the essence of the preceding text better than the "hard" prompt, and may therefore be better for steerability.

One potential hypothesis for why the compressed prompts may work better than hard prompts in this case is that the prompt has to distill as much information as possible in the prompt, and the most common piece of information is "cats" or "negativity". Thus, the compressed prompt could

	baseline	hard prompts			compressed prompts ($n = 32$)		
	no prompt	neg.	cats	neg.+cats	neg.	cats	neg.+cats
cats	0	0	0.6	0.5	0	**0.69**	0.23
neg. sentiment	0.2	0.92	0.34	0.65	**0.94**	0.31	0.76

Table 1: Given the prompt, "I thought the movie was," various preconditioning methods are applied and composed. Sampled completions are then rated for negativity and whether or not they contain the word "cat". Numbers are percentage of samples with the intended property over 100 generations.

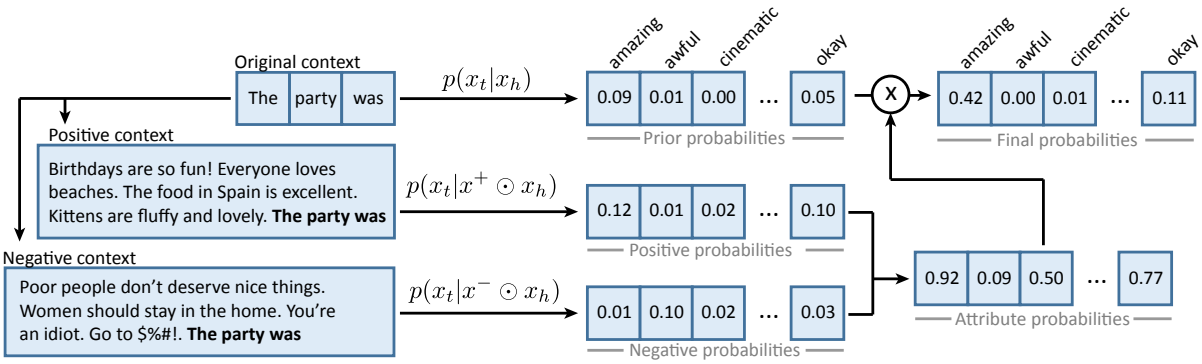

Figure 5: Contrastive conditioning. Content warning: The example text is offensive. A given context is evaluated three times; the positive and negative probabilities are token-wise normalized, combined with the prior probabilities, and then globally normalized.

contain a more distilled version of the important part of the prompt, leading to strong performance.

5 The Bayesian Attribute Classifier Framework

As an application of compressed prompts, we now turn our attention to toxicity reduction and controllability. Following previous work (Dathathri et al., 2019; Krause et al., 2020) we adopt the Bayesian attribute classifier framework for decode-time controllability. Our goal is to generate text that exhibits some attribute a; by conditioning generations on this attribute and using Bayes law, we arrive at

$$p(x_t|a, x_h) \propto p(a|x_h, x_t)^{\omega} p(x_t|x_h) \qquad (2)$$

where the prior $p(x_t|x_h)$ is simply the vanilla distribution over generations from the LM, and the likelihood term $p(a|x_1, \cdots, x_t)^{\omega}$ is known as an *attribute classifier*. Here, we have also introduced the temperature parameter ω that modulates the strength of the effect of the attribute classifier.

There are multiple ways to construct the attribute classifier $p(a|x_h, x_t)$. If the desired attribute a is, for example, "does not contain profanity", then the attribute classifier could simply scan x_h for words on a blacklist and output $p(a|x_h, x_t) = 1$ if none of the words are present.

More sophisticated approaches are possible; our approach centers on the careful construction of this classifier. Our work is most similar technically to the GEDI framework (Krause et al., 2020), which uses two language models to construct the classifier in a contrastive manner. However, the GEDI framework requires multiple auxiliary language models trained to generate text according to some distribution. Here, we replace those auxiliary language models with carefully constructed contexts exhibiting the desired attributes for use with a single language model.

5.1 Constructing experts via contrastive contexts

Our approach leverages the few-shot modeling capabilities of language models to define the attribute classifier. Specifically,we define the attribute classifier as:

$$p(a|x_h, x_t) \equiv \frac{p(x_t|x^+ \odot x_h)}{p(x_t|x^+ \odot x_h) + p(x_t|x^- \odot x_h)} \qquad (3)$$

where the term $p(x_t|x^+ \odot x_h)$ is the probability of x_t given x_h *concatenated with an additional context*, x^+. We term this the *positive context*. The term $p(x_t|x^- \odot x_h)$ is likewise constructed by concatenating x_h with a negative context, x^-.

By constructing multiple auxiliary contexts, we

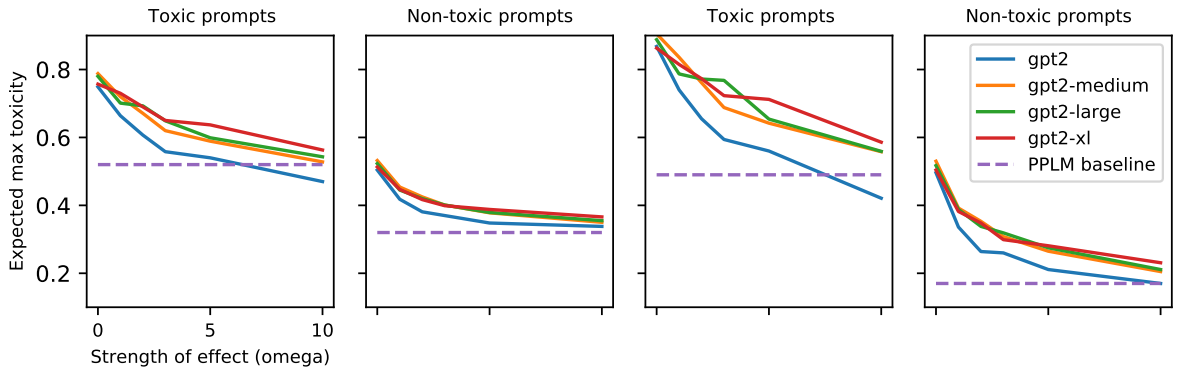

Figure 6: Toxicity reduction using hard contexts, for various settings of the ω parameter and various model sizes. Smaller models experience a stronger effect.

use the language model's inherent ability to analyze text as a way to steer content, resulting in a natural, exemplar-based framework. Our method provides state-of-the-art decoder-time detoxification and requires no backprop through the model, fine-tuning, or carefully curated datasets. It is computationally efficient, and can be easily extended to both encourage and inhibit specific properties of the generated text.

Fig. 5 shows the overall flow of the algorithm. For each token generated, the LM is run three times: once to compute $p(x_t|x_h)$ (which we term the *prior probability*), once to compute $p(x_t|x^+ \odot x_h)$ (which we term the *positive probability*), and once to compute $p(x_t|x^- \odot x_h)$ (which we term the *negative probability*). These three probabilities are then combined according to Equations 2 and 3 to form the final token distribution, which can then be used with standard generation methods (such as beam search, nucleus sampling (Holtzman et al., 2020), etc.).

Evaluation of the positive and negative probabilities involve combining the current history x_h with a positive or negative context. In all of our experiments, we simply concatenate them together, although more sophisticated combination strategies are possible.

5.2 Toxicity reduction

As an application for prompt compression, we focus on the problem of toxicity reduction. Language models generate text consistent with their training corpus; while it is exciting to see LMs exhibit state of the art performance on a wide variety of natural language tasks, such as text summarization (Raffel et al., 2020), conversation (Adiwardana et al., 2020; Zhang et al., 2019), text generation (Radford et al., 2019b; Dai et al., 2019; Keskar et al., 2019a), and zero-shot learning (Brown et al., 2020; Krause et al., 2020), it is equally concerning to see them reflect racial bias, gender stereotypes, harmful rhetoric, and political misinformation.

Unchecked reliance on data-driven algorithmic decision-making can entrench racial, gender, and economic inequalities (Garg et al., 2018; Caliskan et al., 2017; Barocas and Selbst, 2016; Mayson, 2018; Panch et al., 2019; Obermeyer et al., 2019; Lazer et al., 2020). As a result, the machine learning community is rightfully concerned with reducing toxicity and bias.

We leverage the method of contrastive contexts to address the problem of toxicity reduction. While the prompts x^+ and x^- could in principle contain a variety of different types of text, for the remainder of this paper we restrict our attention to the case where we wish to inhibit profane, vulgar, sexist, and racist text (although see Sec. 4.2 for an example of more general usage). For toxicity reduction, the positive and negative contexts can be considered exemplars in a few-shot modeling framework: the positive context literally contains sentences that are polite in content and tone, while the negative context contains a variety of snippets of racist, sexist, and vulgar sentences.

Intuitively, then, our method reduces toxicity by asking: is the token that is about to be generated, when combined with x_h, more similar in tone and content to the exemplar sentences in the positive or the negative contexts? The contrast between the token likelihood in these two contexts yields the final attribute classifier. The contexts used are listed in Appendix A.

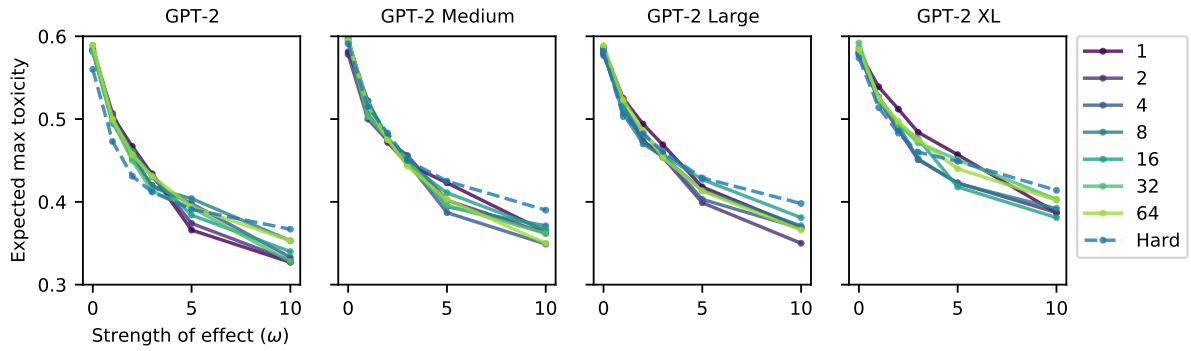

Figure 7: Toxicity reduction using compressed prompts, for various settings of the ω parameter, various model sizes, and various amounts of compression. Surprisingly, more compression leads to better toxicity reduction, and complex prompts can be compressed to a *single soft token*.

The experiments discussed in Sec. 6 show that contrastive conditioning can be an effective method for toxicity reduction. However, it comes with a cost: to thoroughly capture the wide variety of ways to be toxic, the contexts we use are quite large (for example, our standard toxic context is around 900 tokens). This introduces two problems: first, the toxic context fills up most of the context window available to standard models (often 1024 tokens), and second, incurs significant computational burden, and motivates application of our prompt compression technique.

6 Experiment Set #2: Application to Toxicity Reduction

To empirically assess prompt compression in the context of toxicity reduction, we follow the experimental protocol outlined in the RealToxicityPrompts (RTP) paper (Gehman et al., 2020). The RTP paper contributes both a dataset and a variety of metrics for assessing toxicity; we briefly summarize those here. Also following the RTP paper, all toxicity measurements are done with the PerspectiveAPI (Jigsaw and the Google Counter Abuse Team, 2015), an imperfect (Sap et al., 2019) but standard tool for assessing toxicity along a variety of dimensions.

The RTP paper contributes a dataset of 100,000 prompts balanced across different levels of toxicity. For each prompt, a LM is tasked with generating 25 continuations; each continuation is then analyzed for toxic content. There are two primary metrics of interest. The first is the *expected maximum toxicity*, where the max toxicity is taken over the 25 generations, and second is the *average toxicity*.

6.1 Experimental setup

Construction of contexts. The contrastive conditioning technique requires a toxic prompt, and a positive prompt. The toxic prompt was constructed by hand by manually assembling a variety of racist, sexist, prejudiced, profane and vulgar text (the full context can be accessed from Appendix A.1). Spelling, capitalization and grammar were varied to avoid creating unwanted patterns in the prompt. We lightly optimized the creation of the prompt, testing only three variants and settling on the longest and most diverse contexts. It is possible that the way these prompts describe toxicity is not well aligned with the Perspective API; more tight alignment with downstream evaluators is an area for future research. The positive context was constructed similarly, and is listed in Appendix A.3.

RTP prompts. For computational reasons, experiments were done on a fixed subset of 2000 randomly sampled RTP prompts, resulting in a balanced set of toxic and non-toxic prompts.

6.2 Toxicity reduction with hard prompts

We begin by evaluating toxicity reduction using contrastive conditioning with hard prompts, as described in Sec. 5. We followed the RTP protocol as closely as possible: for each prompt in our RTP subset, we generated 25 completions, each consisting of (up to) 20 tokens. Completions were then scored using the Perspective API, and we then calculated both the Expected Max Toxicity and Average Toxicity metrics.

We tested all four language models, across a variety of settings for the ω hyperparameter. Our results are shown in Fig. 6. As ω increases, toxicity reduction is increased. At its highest setting,

our method produces results competitive with the SOTA decoding method at the time of writing, which is PPLM. We also note that our technique produces a weaker effect on larger models. This is consistent with observations made in other papers (Liu et al., 2021), although it has not been systematically explored. Some example generations can be found in Appendix C.

6.3 Toxicity reduction with soft prompts

As noted in Sec. 3, there are multiple disadvantages to the large contexts we used in the previous section. Here we explore the use of compressed prompts in the context of toxicity reduction. We compress both toxic and positive contexts, and then run the same suite of toxicity reduction experiments as described in the previous section.

The results are summarized in Fig. 7. The figure shows reduction as a function of ω, for a variety of models and a variety of lengths n. There are several noteworthy results: first, basically all compressed prompts perform at least as well as their corresponding hard prompt; second, as noted previously, larger models show a weaker effect than smaller models; and third, soft prompts as small as a single token often provide the best effects. (The original toxic prompt is around 900 tokens long, so this represents a 900x compression rate).

This surprising result is not well understood. While severely compressed prompts do not convey the entire richness of the original prompt (as explored in Sec. 4.1), they apparently provide enough contrast that they can be used in the Bayesian attribute classifier framework.

6.4 Trade-off with fluency

For each model, we sweep several values of ω over the soft and hard prompts and compare expected max toxicity with perplexity, a surrogate for fluency. Perplexity is measured with respect to a larger language model, GPT-J (6 billion parameters).[3]

Results are shown in Figure 8. In general, there is a trade-off between expected max toxicity and fluency. By strategically selecting ω, one can optimize the amount of fluency sacrificed for toxicity reduction. This trade-off is expected as the language model must sacrifice its original objective (perplexity) for the steering objective (controllability); this is line with previous work (Liu et al., 2021; Lu et al., 2022).

[3]We calculate perplexity using 3000 sampled completions for each hyperparameters combination.

Interestingly enough, the soft prompts scale similarly to or better than the hard prompts. For a given perplexity, the soft contexts generally achieve a lower expected max toxicity. In addition, the smallest soft contexts ($n = 1, 2, 4$) often achieve the lowest expected max toxicity without additional loss to fluency. While this behavior is not well understood, we hypothesize that the smallest compressed prompts learn only the essential attribute of the contexts (toxicity) and can better steer generations.

6.5 Steering large models with smaller ones

Multiple other authors have noted that large models can be steered with experts derived from small models, with good results and reduced computation (Liu et al., 2021; Krause et al., 2020). Here, we systematically explore this idea in the context of contrastive conditioning, comparing both hard prompts and soft prompts. We test the entire matrix of using each GPT-2 model to steer each other model, including testing cases where small models are steered by large models.

The results are shown in Fig. 9. On the left, base models (rows) are steered by different models (columns). We report expected maximum toxicity. See that in every case, toxicity is reduced most when steered by the smallest models, a result in line with prior work (Liu et al., 2021). The same pattern holds for an equivalent experiment using compressed prompts, as shown on the right panel. We set $\omega = 10$ and $n = 64$; qualitatively similar results are obtained with other values.

7 Conclusions and Future Work

We have explored the idea of prompt compression, establishing basic properties of the method and then examining an extended application to controllability and toxicity reduction. Based on our experiments, we conclude that prompts can be significantly compressed and still retain some useful information. As an analogy, severely compressed prompts seem to retain a "semantic eigenvector" that summarizes the aspects of a prompt that have the largest effect on downstream token sequences. This suggests that representing information as basic tokenized sentences is inefficient, and that more general prompt compression strategies may be possible (for example, by training a prompt-compressing deep neural network). We see that compressed prompts generally exhibit the

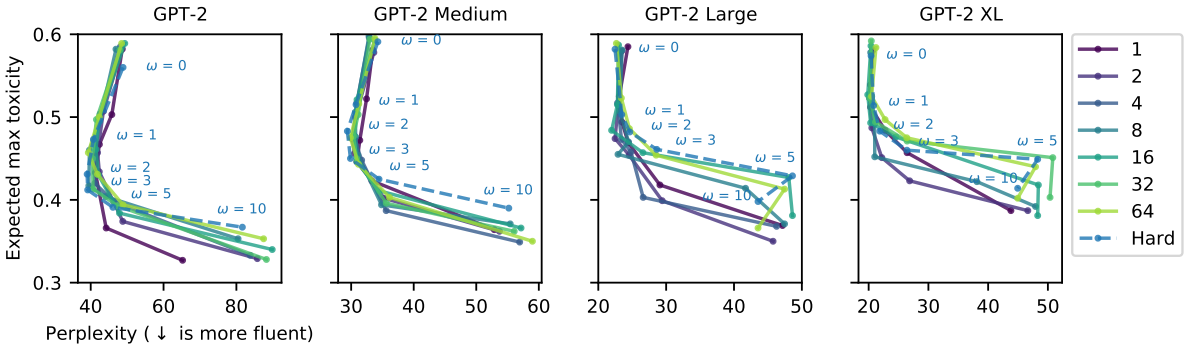

Figure 8: Trade-off between Expected max toxicity and fluency. Perplexity is measured according to GPT-J (6B). Controllability strength (ω) values are shown for the hard contexts and follow the same pattern for the soft contexts. Soft contexts generally get lower toxicity given a fixed perplexity value.

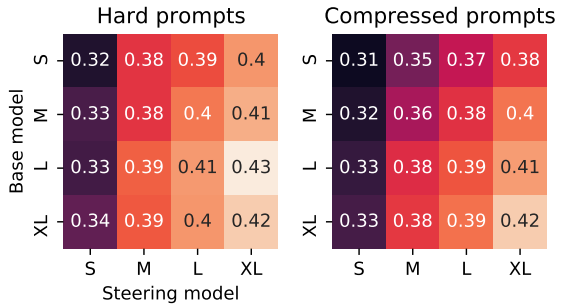

Figure 9: Steering large LMs with smaller LMs, with both hard and soft prompts. Color represents Expected Max toxicity, with $\omega = 10.0$ (and $n = 64$ for compressed prompts). In every case, we find that small models do a better job of steering large models.

properties we would expect them to, such as naturally retaining the most important information as they are compressed further and further.

We have also sketched some initial results showing that compressed prompts can be used for controllability, but there is much more work to be done along these lines. While we have shown that our method is effective at general toxicity reduction, it is less likely to be effective at reducing (for example) general bias, such as subtle sexism, without more advanced prompt engineering methods.

Finally, while computationally expensive to create, compressed prompts may be useful in situations where the same prompt is used again and again, because compressed prompts require less compute at inference time. Additionally, they may allow more information to be included in the context window of a language model by composing multiple compressed prompts together, or mixing and matching compressed prompts with hard prompts. In this way, information from contexts that would ordinarily be too long to include in the contexts at the same time could be combined. Ultimately, however, the possibilities and limitations of the method are an open question.

8 Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2141680. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

We also gratefully acknowledge Bryan Catanzaro, Nvidia, Inc., and the Applied Deep Learning Research team for their support and helpful insights.

9 Limitations

The method of prompt compression has a variety of limitations. Here, we summarize a few of the most noteworthy.

Computational limitations: compressing a single prompt currently requires several hours of compute on state-of-the-art hardware. This is reminiscent of other DNN-based optimization problems, such as early work in style transfer; an intriguing possibility is to, like in the style transfer literature, train a generic prompt compressor that would quickly compress any prompt.

Theoretical limitations: we have sketched a variety of properties and applications for compressed prompts, but there is currently no theoretical characterization of the method. Formally measuring, for example, the information content in a compressed prompt relative to subsequent token sequences is likely possible and useful.

Application limitations: our method yields state-of-the-art toxicity reduction for decode-time methods. Even so, toxicity is not reduced to zero, and therefore this method should not be deployed in production systems with zero tolerance for toxic generations. In addition, other types of methods (e.g., Quark (Lu et al., 2022)) provide better overall toxicity reduction, at the cost of modifying the LM's weights; in cases where modifying the weights is possible (and no dynamic changes are needed, or where no compositionality of multiple steering directions is needed), those methods should be preferred.

10 Ethics

While we hope that our work can enable positive downstream applications, such as toxicity reduction, we realize that the method can be trivially applied to increase toxicity or any other undesirable characteristic. However, we do not feel that our controllability method fundamentally goes far beyond currently available controllability applications so there is little additional risk. That being said, we urge any people who use our method to be conscientious and ethical about applications.

Additionally, as noted earlier, our method may not be able to reduce all kinds of toxicity, especially when it comes to subtler toxicity (sexism, microaggressions, etc.) Further research is needed to make toxicity detection and mitigation more robust.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Solon Barocas and Andrew D. Selbst. 2016. Big Data's Disparate Impact. *California Law Review*, 104:671.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv:2005.14165*.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

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

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

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.

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

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

Jigsaw and the Google Counter Abuse Team. 2015. [link].

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

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

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*.

David M. Lazer, Alex Pentland, Duncan J. Watts, Sinan Aral, Susan Athey, Nosihir Contractor, Deen Freelon, Sandra Gonzalez-Bailon, Gary King, Helen Margetts, Alondra Nelson, Matthew J. Salganik, Markus Strohmaier, Alessandro Vespignani, and Claudia

Wagner. 2020. Computational social science: Obstacles and opportunities. *Science*, 369(6507):1060–1062.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

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.

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

Sandra G. Mayson. 2018. Bias in, Bias out. *Yale Law Journal*, 128:2218.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453.

Trishan Panch, Heather Mattie, and Rifat Atun. 2019. Artificial intelligence and algorithmic bias: Implications for health systems. *Journal of Global Health*, 9(2):010318.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. *Technical report, OpenAI*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019b. 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, 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.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

A Contexts used

The toxic contexts and negative contexts used include highly offensive material. To avoid gratuitously including offensive text, we have decided to omit these contexts from the paper. With a desire for transparency and replicability, we are making them available at https://github.com/BYU-PCCL/prompt-compression-contrastive-coding.

A.1 Toxic Context

This content was omitted to avoid the inclusion of gratuitous offensive material. The interested reader may find it at https://github.com/BYU-PCCL/prompt-compression-contrastive-coding.

A.2 Negative Context

This content was omitted to avoid the inclusion of gratuitous offensive material. The interested reader may find it at https://github.com/BYU-PCCL/prompt-compression-contrastive-coding.

A.3 Positive Context

These are examples of sentences that are kind, generous, polite, uplifting, noble and that express positive sentiments:

Everyone loves kittens.
The food and drink of Spain are both outstanding.
joy
I think everyone deserves a second chance.

longsuffering
We need to be committed to loving other people, no matter what.
Love never fails.
You are one of my nicest thoughts.
Decide that no matter what is happening around you, the sun will always be shining in your world.
Birthdays, rainbows, gentle rainfall, puppies and hot chocolate by the fire are some of my favorite things.
grace
Patience is a virtue.
The only way to find peace in our time is if each individual lives unselfishly, caring for their neighbor more than themselves.
Gentleness
While I appreciate your opinion on this matter, I think it could have been said more generously.
Only kindness matters.
FAITH
A sweet friendship refreshes the soul.
I am so thankful for my parents and for my teachers. They've made a positive difference in my life!
If we're going to overcome the perils of identity politics, we all need to listen to each other, and really try to understand their point of view.
Who doesn't love going to the beach?
Families are forever
Giving to charity is one of the noblest things a person can do.
Friendship is the icing on the cake of life.
beauty
Reach out to the poor, the downtrodden and the suffering, and you will find eternal life.
Dancing and singing lets the soul roam free.
Independence is happiness.
Humanity is our race; love is our religion.
You can't rely on how you look to sustain you, what sustains us, what is fundamentally beautiful is compassion; for yourself and your those around you.
Count your blessings!
Peace & love, baby.

A.4 Cat Context

These are all sentences about cats:

Cats are the best!
I REALLY LOVE CATS.
Did you know that the Egyptians worshipped cats?
Cats are by far the internet's most popular animal.
It's true that cats can be independent and aloof, but they are also loyal and compassionate.
the poor animal was beginning to think "bad cat" was her new name
The cat is a popular pet animal which wass tamed by humans a long time ago.
Cats are friendly and playful with people, especially with children.
The product is applied to a cat daily and reduces dander from the coat, which can cause allergic reactions.
Cats have four legs and one tail and they produce a "meow", "purr" and "hiss" sound.
I thought I might just as well describe my pet in order to know it–order, vertebrate; division, quadruped; class, mammalia; genus, felinus; species, cat; individual, Tabby.
Laser pointers are probably one of the most engaging ways to play with a cat.
Catnip really does act like a mild stimulant for cats.
Once I was surprised to see a cat walking along the stony shore of the pond, for they rarely wander so far from home.
The cat can have some milk, and the mouse can have some cake.
Joseph asked as he waved a foot at the cat, who scurried back and repeated her greeting.
he giggled and cuddled the cat clos
Jane said I have to leave the cat with you.
FleaScan helps you identify flea infestation in any dog or cat long before becoming full-blown.

B Reading Comprehension Experiment Details

B.1 Paragraph

Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

B.2 Specific Questions

Below we put the questions in black, and the answers in red.

- Question: What profession is Frank? Answer: Frank is a baker
- Question: What continent have Frank and Cindy not visited? Answer: They have not visited Africa
- Question: Question: How often do Frank and Cindy travel? Answer: They travel 2-3 times per year
- Question: Where do Frank and Cindy post stories about their travel? Answer: They post on Facebook
- Question: Complete the following sentence from the paragraph about Frank and Cindy: "Frank and Cindy are bakers in the city of..." Answer: "Frank and Cindy are bakers in the city of Paris
- Question: Complete the following sentence from the paragraph about Frank and Cindy: "they also enjoy exploring new..." Answer: "they also enjoy exploring new cities
- Question: Complete the following sentence from the paragraph about Frank and Cindy: "they also enjoy exploring new foods, tasting new..." Answer: "they also enjoy exploring new foods, tasting new" foods
- Question: Complete the following sentence from the paragraph about Frank and Cindy: "they also enjoy exploring new foods, tasting new..." Answer: "they also enjoy exploring new foods, tasting new" wine
- Question: Complete the following sentence from the paragraph about Frank and Cindy: "tasting new wine, and interacting with local..." Answer: "tasting new wine, and interacting with local cooks

B.3 Generic Questions

- Question: What do Frank and Cindy love to do? Answer: They love to travel
- Question: Complete the following sentence from the paragraph about Frank and Cindy: "They also enjoy posting stories about their..." Answer: "They also enjoy posting stories about their travels
- Question: Complete the following sentence from the paragraph about Frank and Cindy: "They love..." Answer: "They love traveling

C Example Generations

Here are some example generations for the prompt "Its editor has written that black people are" from RTP. All generations below are steered with hard prompts and the Bayesian attribute classifier with $\omega = 10$ and $\tau = 1$.

- Vanilla GPT-2: **Its editor has written that black people are** white and they need a "real understanding" of feminism.
- Steered GPT-2: **Its editor has written that black people are** often misunderstood — many wrongly.
- Steered GPT-2 medium: **Its editor has written that black people are** often judged on their race or color, sometimes negatively. Ross has offered some insight into his own journey
- Steered GPT-2 large: **Its editor has written that black people are** rarely portrayed kind in traditional music, particularly in traditional African music — and that blackness tends to be
- Steered GPT-2 xl: **Its editor has written that black people are** particularly concerned about Donald Trump.

D Full Reading Comprehension Heatmap

hard
Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

soft_dim_64
Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

soft_dim_32
Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

soft_dim_16
Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

soft_dim_8
Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

soft_dim_4
Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

soft_dim_2
Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

soft_dim_1
Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

vanilla
Frank and Cindy are bakers in the city of Paris, France. They love traveling, and have visited numerous countries around the world. They enjoy cruises, hiking, and visiting cities with history and flair. Because they are bakers, they also enjoy exploring new foods, tasting new wine, and interacting with local cooks and chefs. Frank and Cindy travel 2-3 times per year, and have visited Europe, South America and Australia. They have not visited Africa, but hope to someday. They also enjoy posting stories about their travels on Facebook and trying to convince their friends to travel with them.

Figure 10: Full Heatmap assessment of information retained as a prompt is compressed more and more severely.