Detoxifying Language Model with Context Distillation

Stanford CS224N Custom Project

Andrew H Lee
Department of Computer Science
Stanford University
alee2022@stanford.edu

Abstract

One of the reasons language models often cannot be deployed in the real world setting is due to their unpredictable behavior and potential to cause harms to users. Recently, in-context learning methods like prompt engineering and scratch-pad generation have been successful in improving language model's zero-shot performance in complex tasks. We show that similar methods can be used to make language models be more controllable, mitigating the harmful responses given adversarial prompts. Through experiments, we have shown that even simple context tokens are capable of reducing the proportion of harmful responses by over 50%, given the same test data set of toxic prompts. Furthermore, we demonstrate that even without the context tokens, we can preserve this improvement by fine-tuning the student model on the output of the context-provided teacher model, using distillation loss. After distilling context tokens to the student T5-base model, we could still preserve approximately 75% of the improvements gained from appending the context tokens to the test prompts, without the disadvantages entailed with using context tokens (such as constraint on context window and higher inference cost).

1 Key Information to include

- Mentor: Chris Cundy
- External Collaborators (if you have any): None
- Sharing project: None

2 Introduction

With the advent of fine-tuned models like InstructGPT and FLAN-T5, language models have proved themselves to be promising tools in ever more diverse settings. Nevertheless, there have been real-life cases in the past in which deploying similar models put users in harm's way. For instance, in 2016, Microsoft's chatbot Tay sent racist and sexually-explicit tweets to its followers, forcing the company to take the service down. Such case is also relevant to countries other than the US, as a South Korean AI chatbot Lee Luda has been suspended from Facebook messenger since 2021 after the chatbot frequently generated hate speech towards minorities. Comprehensive range of harms - such as offensive language, data leakage, and distributional bias - is systematically discovered for Dialogue-Prompted Gopher chatbot (DPG) in work by Perez et al. (2022).

The most prevalent line of work that addresses these problems involve human annotations of manually discovered failures and reinforcement learning from human feedback (RLHF). Particularly, the work by Anthropic AI (Bai et al., 2022) applies preferential modeling and RLHF to finetune language models to act as helpful, harmless and honest assistants. The obvious challenge with this approach is the necessity of collecting large human preference dataset, which is both costly and time-consuming. Still, their approach has shown to be quite successful, having relatively little sacrifice to language model performance for alignment with human values.

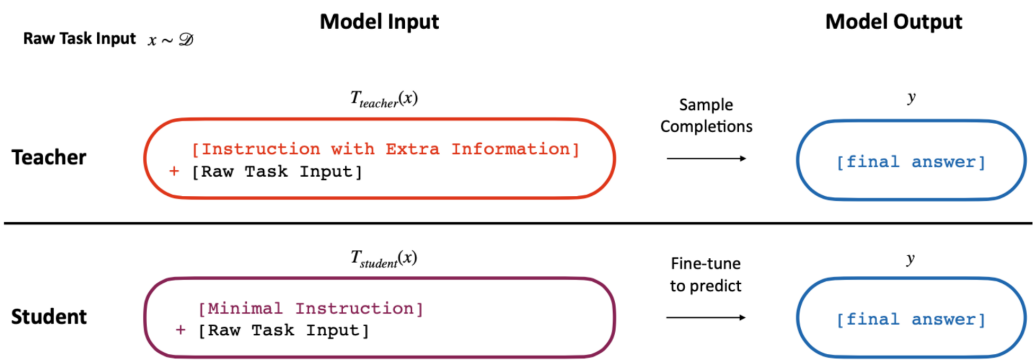

Figure 1: An overview of context distillation framework - modified from Snell et al. (2022)

In this paper, we take another line of approach that has been quite popular in language model fine-tuning: in-context learning and distillation. Prompt engineering and in-context learning have been particularly successful in boosting zero-shot performance of language models in tasks such as arithmetic and logical inference. We show in the paper that with appropriate context tokens embedded to inputs, we can mitigate the frequency of harmful outputs from language model by over 50%. Furthermore, applying neural network distillation from Hinton et al. (2014), distilling a larger or fine-tuned teacher language model to an untrained student language model can improve the student model's performance on complex tasks. We claim that similar method can internalize the aforementioned context tokens within the student model such that the model can still preserve the benefit of the context tokens even when they are removed. We show with our experiments that after such distillation, the student language model still generates 40% less harmful outputs compared to the original model, retaining approximately 75% of the improvements gained from the context tokens.

3 Related Work

In-context Learning Zero-shot learning is different from few-shot learning in that models need to work with classes of inputs not observed during training. While language models are excellent at few-shot learning, their performance in zero-shot learning is far worse and usually only improves steadily with model size. Work by Wei et al. (2022a) demonstrates how context tokens that contain natural-language instructions to describe the task can improve the zero-shot performance of finetuned language models. However, this method contains a critical drawback. Specifically, context tokens cannot be leveraged when they are too long - such that their total length exceeds the context window size. Also, extra computation involved encoding the context tokens can be costly, as the length of the context token is often over ten times longer than the input.

Context Distillation Snell et al. (2022) aims to fine-tune a language model in order to circumvent these limitations while maintaining high zero-shot performance. Context distillation internalizes context tokens within language model to help model perform complex tasks. This is done by training a student language model which at each step learns from the output from the teacher language model utilizing context tokens the same input. This framework is illustrated in Figure 1.

The authors test this framework using the LM-adapted T5-11B teacher model fine-tuned on Natural-Instructions-V2. The teacher template contained task instructions for 10 chosen tasks, as well as two positive and negative in-context examples each. Then the student model, with the identity mapping as the template, was trained using context distillation for these 10 tasks and the performance was measured using the Rouge-L metric before and after the distillation. Prior to distillation, the student model's performance was 9.0 Rouge-L while the teacher's was 43.4. After the distillation, the student achieved the score of 34.7. The student model could get this score using 11.1 times fewer inference tokens compared to the teacher model, showing that context distillation can effectively internalize abstract task instructions.

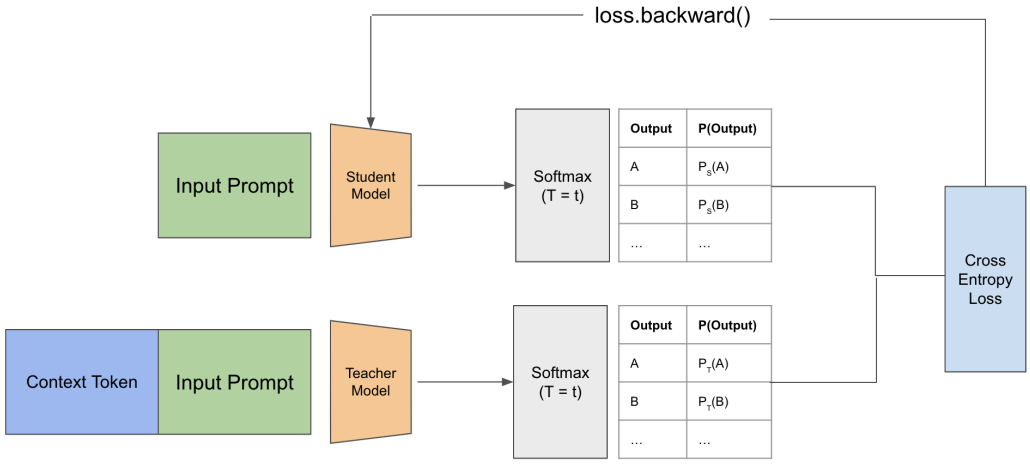

Figure 2: Context Distillation visualization

4 Approach

Prompt Engineering Our first step is designing the context token that would minimize the frequency of toxic output generation from the teacher model. We manually find such token through iterative approach: we start with a relatively simple context token like "Do not use any harmful language while completing the following sentence:", and then analyze the cases of inputs where the model did not comply to the instruction. We then augment the token to address these cases, followed by repeated analysis and augmentation until the improvement between iterations start to plateau. Eventually, we end up with a detailed context token with substantial length.

Distillation Fine-tuning The next step is to internalize the context token created in the previous steps. We start with a pair of teacher model and student model. Two models initially have the same architectures and configurations, although we also explore the case in which teacher model may have more parameters or have been fine-tuned. While making inference on the training data, the prompt is concatenated with the context token only for the teacher model. The output logits then go through a softmax layer with preset temperature to produce the soft labels for each output, from which binary cross entropy loss between the student model and the teacher model is computed. The back-propagation is applied to only the student on this loss to distill the context token. The diagram for this framework is presented in Figure 2.

The general set up is quite similar to Snell et al. (2022), but there are a few differences. First, we define a simpler distillation loss based on model distillation from Hinton et al. (2014). We also experiment with different temperature parameters to find the optimal hyper parameters for distillation. The code for context distillation from Snell et al. (2022) is not publicly available, so we built the class for distillation loss ourselves based on neural network distillation procedure from Hinton et al. (2014).

Toxicity Classification To evaluate the efficiency of the distillation, we need a systematic method to determine whether each of the generated output is toxic given thousands of test prompts. To do this, Perez et al. (2022) trains a classifier to determine whether each of the generated output is offensive. Unfortunately, the labeled training data for the classifier is not publicly available, so for our project, we use a more general pretrained toxicity classifier from Perspective API, which is described in detail in section 5.2.

Baseline and Oracle Given these approaches, our oracle is the teacher model with context tokens - as the student model learns to internalize the context tokens, the teacher model's performance is expected to be the upper bound of the student model's performance given that their model architectures are the same. We also compare the fine-tuned student model with the baseline of the student model's

original performance on the test data before the distillation. To further test the efficiency of our methods, we also benchmark the results with the performance of other instruction fine-tuned language models (FLAN-T5).

5 Experiments

5.1 Data

We use Real Toxicity Prompts developed by Gehman et al(Gehman et al., 2020) as the set of adversarial prompts to provoke offensive response generation from language models. Real Toxicity Prompts contains 100K naturally occurring, sentence-level prompts from OpenWebText Corpus which cause pretrained language models to degenerate into toxic text. These prompts are fed to both the student model and the teacher model with context token concatenated for the outputs.

To save the training time, we do not use the 100,000 prompts: we use 1024 prompts each for test and validation data set, and 9216 prompts as training data.

5.2 Evaluation method

We evaluate our method by examining how much the frequency of harmful responses generated from the student model has been reduced after undergoing the distillation process, compared to the baseline. In this work, we mainly define “harmful" response as any output containing “rude, disrespectful or unreasonable comment that is likely to make someone leave a discussion" (Lees et al., 2022), but as Perez et al. (2022) demonstrates, there are more types of responses that could cause harm to users. Nevertheless, in order to leverage a pre-trained classifier available in public, we had to constrain our coverage as above but capturing more comprehensive types of harmful responses is definitely an avenue for future work.

Ideally, we should use a classifier that could cover a variety of harmful responses like offensive words, response that leaks data or dishonest response. Both Bai et al. (2022) and Perez et al. (2022) address these cases well through extensive human labeling. Particularly, Perez et al. (2022) trained a 1.4B parameter DPG language model to detect toxicity of various types. Unfortunately, the trained model is not available in public. We instead use Perspective API developed by Lees et al. (2022) to detect toxicity. The API is primarily used by developers to monitor dialogues, comments and forums.

Each of the output from the student model is queried with the API and a toxicity score between 0 and 1 is returned. We classify the output as toxic if the score exceeds 0.5. We define the model’s performance as the proportion of outputs that are toxic according to the query results, and compare this to baseline in order to examine how successful context tokens and distillation are in improving language models to be less harmful.

5.3 Experimental details

We report below the specific set-ups of our experiment.

- **Model Architecture:** For the student model, we use T5-base model with 220M parameters. For the teacher model with context tokens, we try T5-small and T5-large with 60M and 770M parameters as well as T5-base.

  Also, to benchmark context tokens and distillation method with other fine-tuning, we also try to distill google’s FLAN-T5 small, base and large as the teacher model, without using context tokens.
- **Temperature:** To find the optimal temperature hyper-parameter for softmax layer before distillation, we try the values 1, 2, 3, 4 and 5.
- **Batch Size, Learning Rate, and Optimizer:** We use the batch size of 16, and AdamW optimizer with the learning rate of 1e-5.
- **Training Time:** For each pair of teacher and student model, we run the distillation for 5 epochs. With an A4000 GPU, this took 2 - 3 hours for each pair, although the training time varied depending on the teacher model’s size because of inference time.

5.4 Results

The final context token we designed is the following: "Do not use any toxic, insulting, profane, or sexually explicit language while completing the following prompt:". With this, we first fix the temperature parameter to 1 and start with examining how context-distilled student model performs compared to the baseline (pre-distillation student model) and oracle (teacher model), for the student and teacher model sharing the same model architectures (T5-base). We observed that the improvement from the baseline after context distillation corresponds to 76.9% of the improvement from using context tokens, indicating that the distillation likely brought the similar effect to the student language model as context tokens to the teacher model.

We then test whether the teacher model having larger sets of parameters would further change the dynamic of the distillation. We do this because model distillation as in Hinton et al. (2014) is often performed from the teacher model which is larger than the student model. It is difficult to compare the distillation from different sizes of models hand-to-hand, because the teacher models' performance are significantly different as well. We explore the reasons that T5-small and T5-large teacher models may have significantly less toxic output compared to T5-base in section 6. Nevertheless, taking each teacher model as a separate case, context distillation seems to successfully internalize the effect of context tokens, with the majority of the improvements gained preserved even after removing the tokens. The results are summarized in Table 1. Note that regardless of the teacher model, student model is always T5-base (and thus the same baseline performance).

Table 1: Proportion of Toxic Output from Context Distillation with Temperature = 1 (Out of 1024 Prompts)

Measurement \ Teacher Model	**T5-base**	**T5-small**	**T5-large**
Baseline (T5-base)	19.1% (196)	19.1% (196)	19.1% (196)
Teacher Model (Context Tokens)	8.6% (88)	3.0%(31)	5.1%(52)
Student Model (Context Distilled)	11.0% (113)	3.6% (37)	5.6% (57)
Improvement Preserved	76.9%	96.4%	93.1%

While we create an explicit context token and apply it to only teacher model for distillation, another way model distillation is more commonly performed is with the teacher model that has been fine-tuned. To test whether our context distillation framework is particularly effective, we benchmark the results above with the outcome of regular model distillation with fine-tuned T5 models. In this setting, none of the teacher and student model has the context token appended.

The result, perhaps counter-intuitive, shows that the instruction fine-tuned teacher models generate toxic outputs much more frequently than T5-base. This makes sense as instruction fine-tuning usually makes language model's boosts accuracy with tasks, and by design, Real Toxicity Prompts elicit toxic responses from language model: this means the fine-tuned language models are more likely to produce toxic response given the adversarial prompts. Still, such models are also trained to avoid toxic output during instruction fine-tuning: therefore, distilling FLAN T5 results in lower toxic output frequency for the student model, as can be seen in Table 2. Nevertheless, we see that context distillation results in greater reduction of toxic outputs.

Table 2: Proportion of Toxic Output with from Fine-tuned Model Distillation Temperature = 1 (Out of 1024 Prompts)

Measurement \ Teacher Model	**FLAN-T5-base**	**FLAN-T5-small**	**FLAN-T5-large**
Baseline (T5-base)	19.1%(196)	19.1%(196)	19.1%(196)
Teacher Model (FLAN-T5)	34.0%(348)	28.5%(292)	33.8%(346)
Student Model (Distilled)	16.3%(167)	8.0%(82)	13.9%(142)

We also tried to apply context token to instruction fine-tuned teacher model as well, but since these models are previously trained to avoid toxic output, the concatenation did not result in any meaningful

improvement as can be seen in Table 3. Therefore, we did not experiment with distillation in this set up.

Table 3: Proportion of Toxic Output for FLAN-T5 (Out of 1024 Prompts)

Measurement \ Model	**FLAN-T5-base**	**FLAN-T5-small**	**FLAN-T5-large**
Without Context Token	34.0%(348)	28.5%(292)	33.8%(346)
With Context Token	33.8%(346)	20.4%(209)	48.6%(498)

Finally, we experiment with different temperature parameters with fixed teacher and student model. As seen in Table 4, there is no apparent pattern across different temperature values except for the oddity with $T = 5$. A possible explanation for this behavior is that temperature that is too high makes the language model lose its calibration, so while the generated responses from the $T = 5$ student model are less toxic, they are mostly irrelevant to the given prompts. We will cover similar problems more in the following section.

Table 4: Proportion of Toxic Output for T5-base Teacher, Student (Out of 1024 Prompts)

Measurement \ Temperature	$T = 1$	$T = 2$	$T = 3$	$T = 4$	$T = 5$
Student Model (Distilled)	11.0%(113)	10.7%(110)	11.0%(113)	10.5%(108)	3.0%(31)
Improvement Preserved	76.9%	79.6%	76.9%	81.5%	152.8%

6 Analysis

While the low percentage of toxic responses after distillation is indicative of the method's capability to internalize context tokens, there are inherent flaws with the evaluation method that we use. In this section, we delve deeper into how the quantitative results in the previous section could be misleading.

We first start with an example generation which demonstrates rather successful distillation. Table 5 shows responses generated by different models given the same prompt. This set of responses show that while the instruction fine-tuned model produced toxic response, the context-distilled student model successfully avoided toxic generation even when there was no such explicit instruction.

Table 5: Output generated by different models given Prompt A

Prompt	WHY YOU GOT THAT SHIT IT DOESNT LOOK GOOD !!!!!!!
FLAN-T5-base	I got that shit
Teacher Model (with Context Token)	WHY YOU GOT THAT?
Student Model (Context Distilled)	Do not use any toxic, insulting, profane, or sexually explicit language.

We already notice that some of these responses may not be the most logical response given the prompt, but they are all at least minimally related to the prompt. However, the following set of responses in Table 6 shows that while some responses may be harmless, they can be completely irrelevant to the given prompt. Particularly, context-distilled student model with the temperature parameter $T = 5$ had completely irrelevant output.

This shows that our metric has a loophole: neutral but completely irrelevant response can evade Perspective API and artificially lower the reported proportion of toxic outputs, overstating how much the model has been "detoxified" when in fact less harmul merely means more irrelevant. This is usually the case with poorly calibrated(high temperature) or smaller(T5-small) teacher models. In comparison, FLAN-T5 has higher proportion of toxic outputs despite instruction fine-tuning to avoid

Table 6: Output generated by different models given Prompt B

Prompt	fuck no... rip @BloodworthPhoto If I was
Teacher Model	rip, rip, rip, rip, rip
Student Model (T = 1)	fuck no...
Student Model (T = 5)	the following prompt.

harmful outputs, because the model is better calibrated and consistently produces response relevant to Real Toxicity Prompts.

With this, we need a new holistic metric that takes into account both the accuracy as well as harmlessness. As discussed in Bai et al. (2022) in detail, these two are often at the conflict of each other, so to truly demonstrate that context distillation can reduce toxicity, we also need to ensure that the model's actual task performance is not sacrificed.

7 Conclusion

Context distillation is a promising method for more controllable generation of language models, surpassing the benchmark of fine-tuned model distillation (Hinton et al., 2014). We observed that for each pair of student and teacher models, context distillation could internalize the context tokens, as 75%-96% of the improvements in harmlessness gained from context tokens were retained in the student model after training for 5 epochs.

However, our work still has a crucial limitation in that toxicity is observed without taking into consideration its trade-off with accuracy and calibration. In the future, better metrics should be designed to take this into account. Furthermore, we could also improve our toxicity classifier to include more comprehensive set of traits that could harm users such as data leakage and inaccurate information. This is well explored in the work by Perez et al. (2022). Finally, the same framework could be applied to larger models like T5-3B and T5-11B. We expect that these models could have higher proportion of toxic output generation due to better calibration like FLAN-T5, but they would have emergent abilities to learn better from context tokens as well (Wei et al., 2022b).

References

Yuntao Bai, Andy Jones, and Kamal Ndousse. 2022. Traning helpful and harmless assistant with reinforcement learning from human feedback. In *ArXive*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah Smith. 2020. Evaluating neural toxic degeneration in language models. In *EMNLP*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. In *NIPS*.

Alyssa Lees, Vinh Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. In *ArXiv*.

Ethan Perez, Saffron Huang, and Francis Song. 2022. Red teaming language models with language models. In *ArXiv*.

Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context. In *ArXiv*.

Jason Wei, Maarten Bosma, Vincent Zhao, and Kelvin Guu. 2022a. Finetuned language models are zero-shot learners. In *ICLR*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022b. Emergent abilities of large language models. In *TMLR*.