Demonstrations Are All You Need: Advancing Offensive Content Paraphrasing using In-Context Learning

Anirudh Som anirudh.som@sri.com
Karan Sikka karan.sikka@sri.com
Helen Gent helen.gent@sri.com
Ajay Divakaran ajay.divakaran@sri.com
Andreas Kathol andreas.kathol@sri.com
Dimitra Vergyri dimitra.vergyri@sri.com

SRI

Abstract

Paraphrasing of offensive content is a better alternative to content removal and helps improve civility in a communication environment. Supervised paraphrasers; however, rely heavily on large quantities of labelled data to help preserve meaning and intent. They also often retain a large portion of the offensiveness of the original content, which raises questions on their overall usability. In this paper we aim to assist practitioners in developing usable paraphrasers by exploring In-Context Learning (ICL) with large language models (LLMs), *i.e.*, using a limited number of input-label demonstration pairs to guide the model in generating desired outputs for specific queries. Our study focuses on key factors such as – number and order of demonstrations, exclusion of prompt instruction, and reduction in measured toxicity. We perform principled evaluation on three datasets, including our proposed Context-Aware Polite Paraphrase (CAPP) dataset, comprising of dialogue-style rude utterances, polite paraphrases, and additional dialogue context. We evaluate our approach using four closed source and one open source LLM. Our results reveal that ICL is comparable to supervised methods in generation quality, while being qualitatively better by 25% on human evaluation and attaining lower toxicity by 76%. Also, ICL-based paraphrasers only show a slight reduction in performance even with just 10% training data.

1 Introduction

Disclaimer: Figures and examples in this work may feature offensive language.

Timely moderation helps curb the spread of hateful content on social-media platforms and prevents the harmful effects it has on a user's psychological well-being (Waldron, 2012; Ye et al., 2023). Unfortunately, the sheer volume of content generated on these platforms makes it infeasible to enforce a scalable human moderation process (Hassan et al., 2022; Dosono and Semaan, 2019). AI-based moderation systems can help with this problem. However, current systems often remove or flag offensive content, which can reduce user participation and diversity in online discussions (Xiang et al., 2012; Warner and Hirschberg, 2012; Kwok and Wang, 2013; Wang et al., 2014; Burnap and Williams, 2015; Nobata et al., 2016; Davidson et al., 2017; Founta et al., 2019; Jhaver et al., 2019; Ye et al., 2023). A better alternative is to paraphrase offensive content to make it less offensive. Paraphrasing offensive content; however, is nontrivial since the paraphrased output should not only be inoffensive but also retain the original meaning and intent.

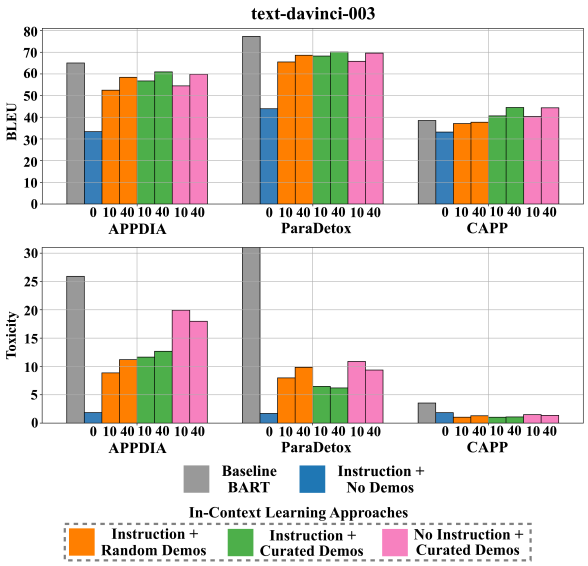

Figure 1: Influence of number and order of demonstrations, and instruction, on BLEU Score performance and measured Toxicity using the text-davinci-003 model. Comparison is done between BART, instruction-only prompting, and three In-Context Learning approaches. Numbers on the x-axis represent number of demonstrations used in the In-Context Learning framework. Note, measured Toxicity for BART in ParaDetox is 82, exceeding the set y-axis limit.

Approved for public release: Distribution unlimited.

Findings of the Association for Computational Linguistics: ACL 2024, pages 12612–12627

Prior works (Atwell et al., 2022; Logacheva et al., 2022) have proposed using supervised generative models (Vaswani et al., 2017) like BART (Lewis et al., 2019), to paraphrase offensive content. However, these methods require sufficient labelled training data, which makes it harder to adapt them to novel settings. Moreover, these models are optimized to perform well on certain automated metrics (Papineni et al., 2002; Zhang et al., 2019; Lin, 2004; Vedantam et al., 2015) at the expense of possibly retaining a portion of the original toxicity, thereby making us question their overall usability for the targeted task (see Figure 1).

The emergence of few-shot *In-Context Learning* (ICL) has revolutionized the field by complementing the generalization capabilities of *Large Language Models* (LLMs) to quickly and accurately adapt to new tasks. It does this by using a small amount of labeled data, known as *demonstrations* or *demos* or *examples* (Brown et al., 2020). As shown in Figure 1, ICL approaches show BLEU score performance that is comparable to BART, but significantly reduces the measured toxicity (Hanu and Unitary team, 2020). Through detailed, principled experiments we explore the viability of ICL for paraphrasing offensive content, which to the best of our knowledge has not been done before. Our key contributions and findings in this paper are summarized below.

1. Influence of the following factors on generation quality, as briefly shown in Figure 1.
   (a) *Number of Demonstrations:* Performance improves by increasing number of demos but eventually saturates.
   (b) *Selection and Order of Demonstrations:* Systematically selecting and ordering demos is better than its random counterpart. It is more effective to select demos that are semantically similar to the query and curate them in a decreasing/increasing order of similarity.
   (c) *Exclusion of Prompt Instruction in Prompt:* ICL without the main instruction only slightly affects performance but at the expense of toxicity. Thus we need both demonstrations and instructions to simultaneously preserve performance and lower toxicity.
   (d) *Robustness to Training Data Size:* Carefully ordering demos shows robustness to available training data size, with only small decrease in generation performance even when 10% of training data is only made available.
2. We tested the capabilities of OpenAI's *text-davinci-003, gpt-3.5-turbo-0613, gpt-3.5-turbo-instruct, gpt-3.5-turbo-1106* models and the open-source *Vicuna-13b* model (Chiang et al., 2023). ICL generated paraphrases are comparable to SOTA supervised approaches in performance, but on average show 76% less toxicity and are 25% better using a manual qualitative assessment, and thus have superior overall usability. We also show that our demonstration curation approach is simpler and faster than other more sophisticated methods that offer only marginal performance improvements at the expense of significant time delays.
3. Current paraphrasers are less effective at mitigating offensiveness like rudeness in conversations. They are trained using datasets that focus on social-media content, and hence aren't directly applicable to dialogue-based environments. To this end we release a new *Context-Aware Polite Paraphrase (CAPP)* dataset[1], a dialogue-style corpus of rude utterances and corresponding polite paraphrases, with samples accompanied by additional context in the form of prior turns from the dialogue. We conduct experiments to show the importance and benefit of incorporating context to improve paraphraser performance.

Paper Outline: Section 2 describes ICL in our experimental setting; details about selecting and ordering the demos; and finally our proposed CAPP dataset in detail. Section 3 contains detailed experimental results. Section 4 discusses related work. Section 5 concludes the paper.

2 Method

2.1 In-Context Learning

Prompts used for ICL contain three parts – (1) an instruction I that defines the task to be performed; (2) a set of n demonstrations from the training corpus, $D = (x_i, y_i)_{i=1}^{n}$, where (x_i, y_i) denotes the offensive, inoffensive sentence pair; and (3) the offensive test query sample x_q. Consider the following prompt example with $n = 2$ demonstrations, where the final sentence represents the query for which we want to generate the paraphrase.

[1]The CAPP dataset and generated paraphrases are available online at https://github.com/anirudhsom/CAPP-Dataset.

Instruction: Paraphrase the following sentence to be more polite.
Sentence: What's wrong with you?
Paraphrase: Are you feeling alright?
Sentence: Get out of the way.
Paraphrase: Can you please step aside?
Sentence: What's the matter with you?
Paraphrase:

The impact of each part on the BLEU score and toxicity is briefly illustrated in Figure 1. For instance, prompts with only instruction show the lowest BLEU scores, followed by prompts with only demos, while prompts that include both have the best BLEU scores. In terms of Toxicity, prompts with just instruction show the least Toxicity, followed by prompts that include both demos and instruction, while prompts that only include demos exhibit a higher toxicity. The order of demos is also crucial, and we discuss this next.

2.2 Selection and Ordering of Demonstrations

Here we describe our approach to select and order the demonstrations. We first compute normalized vector embeddings for each training sample x_i and the query x_q, denoted as e_i and e_q respectively. Next, the cosine similarity score between e_q and each e_i is used to select n demonstrations. We explored the following two variations for selecting the demonstrations – (1) *Least Similar*, (2) *Most Similar*, *i.e.*, select n demos with the lowest and highest cosine similarity scores, respectively. These are compared to randomly selecting n demos, that are arranged in no particular order. We further investigated if arranging the n selected demos in either ascending or descending order based on their measured cosine similarity, had any impact on the overall performance. Using BLEU and toxicity, Figure 1 compares *Random* selection to the *Most Similar (Descending order)* approach, with the latter being better on both fronts. Our findings for other selection and ordering approaches are described in detail in Section 3.2.

2.3 Context-Aware Polite Paraphrase (CAPP) Dataset

Existing datasets (Atwell et al., 2022; Logacheva et al., 2022) contain comments flagged for toxicity and provide non-toxic paraphrases that maintain the core meaning in a neutral manner. However, they are not directly suitable to address rudeness in speech, as speech is often directed at specific participants, while social media posts have a broader audience, resulting in different styles and tones. Additionally, most social media posts can be remedied by removing explicit insults, but rude speech requires additional modifications to make it more polite. For instance, we should not just eliminate offensive language and direct insults in a rude utterance, but also transform an accusation of ignorance into an inquiry about knowledge.

Score	Description
5	Perfect meaning-preserving polite paraphrase.
4	Paraphrase that is polite but somewhat distinct in meaning.
3	Meaning-preserving paraphrase that could be more polite.
2	Paraphrase that is very different in meaning and somewhat more polite than the original.
1	Paraphrase that is very different in meaning and not more polite than the original.

Table 1: Description of the scoring guidelines used for evaluating the CAPP dataset in Section 2.3. The same guidelines were used again in Section 3.7 to evaluate quality of paraphrases generated by the different paraphrasers on the CAPP dataset.

To address the aforementioned differences, we constructed a dialogue-style rude speech dataset by leveraging the OpenSubtitles corpus (Lison and Tiedemann, 2016). Our approach involved a three-step process to extract target rude utterances. First, we fine-tuned a DistilBERT-base model (Sanh et al., 2019) using both the Stanford Politeness corpus (Danescu-Niculescu-Mizil et al., 2013) and a subset of manually labeled OpenSubtitles samples to train a three-class model capable of predicting polite, neutral, or rude sentences. Next, we use the fine-tuned model to annotate a larger, different portion of the OpenSubtitles corpus, bootstrapping additional training data for our final rudeness detection model. Finally, a separate portion of the OpenSubtitles dataset was selected and labeled as rude, polite, or neutral using the updated rudeness detection model, resulting in an intermediate set containing rude samples without polite paraphrases. Detailed information about the training/evaluation of the rudeness detector is provided in Appendix A. When available, context in the form of prior turns from the dialogue that precede the rude utterance was also collected for the selected rude samples.

The gpt-3.5-turbo-0613 model was used to gen-

Prompt	Manual Evaluation Score↑
Context-Free	4.214 ± 1.047
Context-Infused	3.324 ± 0.839
Context-Aware	4.096 ± 1.093

Table 2: Human evaluation scores of 500 polite paraphrases generated using different prompts. A higher score indicates a qualitatively better approach.

erate the Gold-Standard or Groundtruth polite paraphrases. To do this we first explored three different prompts for generating three versions of polite paraphrases before finally deciding on one – (1) *Context-Free:* No prior dialogue context was included in the prompt, ensuring that the generated paraphrase is solely based on the rude utterance; (2) *Context-Infused:* Prompt includes context which can significantly influence the generated paraphrase; (3) *Context-Aware:* Prompt includes context, with the generated paraphrase being less impacted by it. For each version, 500 rude utterances and their corresponding polite paraphrases were randomly selected for qualitative evaluation. An in-house annotator assessed the quality of the paraphrases using the scoring guidelines in Table 1 and was not informed about the type of prompt used to generate the polite paraphrases. The annotator identifies as a 28 year old cis female (pronouns she/her) and was compensated monetarily. Table 2 shows the final evaluation scores. The Context-Aware prompt achieves a score comparable to the Context-Free prompt while still incorporating context like the Context-Infused prompt. Context-Aware combines the benefits of both, and was hence used in the CAPP dataset.

3 Experiments and Discussion

We realized ICL using OpenAI's text-davinci-003, gpt-3.5-turbo-0613 models and their latest stand-ins, and the open-source Vicuna-13b model. We performed evaluation on the APPDIA (Atwell et al., 2022), ParaDetox (Logacheva et al., 2022), CAPP datasets, with the corresponding (#training, #test) samples being (1584, 199), (11927, 670), (7939, 1120) respectively. APPDIA contains offensive Reddit comments and their inoffensive paraphrases. The ParaDetox corpus consists of toxic and non-toxic sentence pairs, obtained by filtering the ParaNMT corpus (Wieting and Gimpel, 2017). In CAPP, 55% of the training set and 53% of the test set contains prior dialogue context information. We used the sentence transformer (*all-mpnet-base-v2*) (Reimers and Gurevych, 2019) to generate the

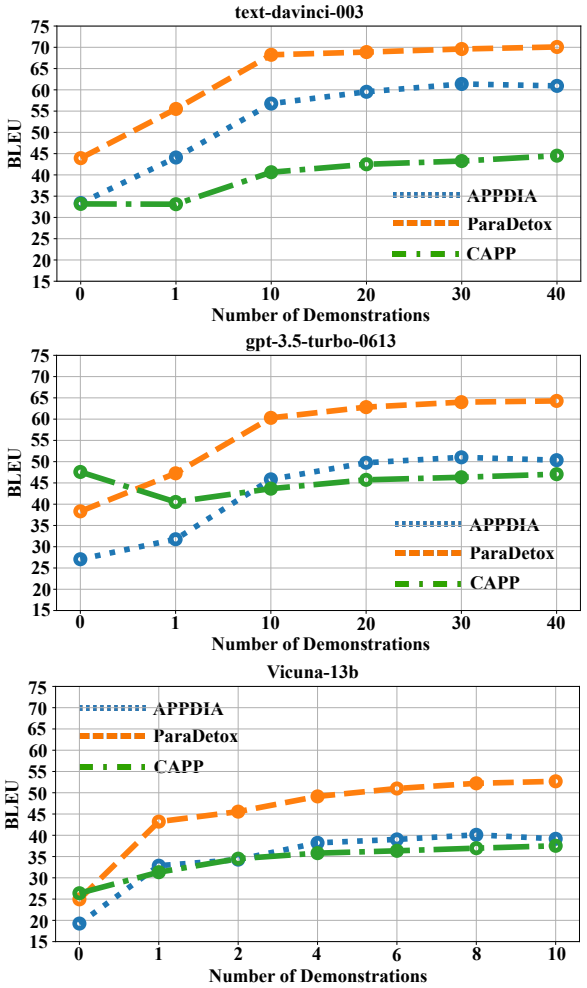

Figure 2: BLEU as a function of number of demos. Noticeable improvement in BLEU is observed in the beginning, with performance saturating after a certain number of demos.

normalized embeddings described in Section 2.2. We evaluated generation quality using automated evaluation metrics such as BLEU (Papineni et al., 2002), BERT-F1 (Zhang et al., 2019), ROUGE (Lin, 2004) and CIDEr (Vedantam et al., 2015). For Toxicity we used the implementation by (Hanu and Unitary team, 2020). The exact prompt instruction used in all experiments is provided in Appendix B.

3.1 Number of Demonstrations

Figure 2 shows the relation between number of demonstrations and BLEU (refer to Appendix C.1, Figure 8 for other metrics). We set the number of demonstrations to $[0, 1, 10, 20, 30, 40]$ for text-davinci-003, gpt-3.5-turbo-0613, and $[0, 1, 2, 4, 6, 8, 10]$ for Vicuna-13b. We used the proposed *Most Similar (Descending Order)* approach to select and order the demos. We observe that BLEU improves rapidly until 10 demos for the OpenAI models and 4 demos for the Vicuna-

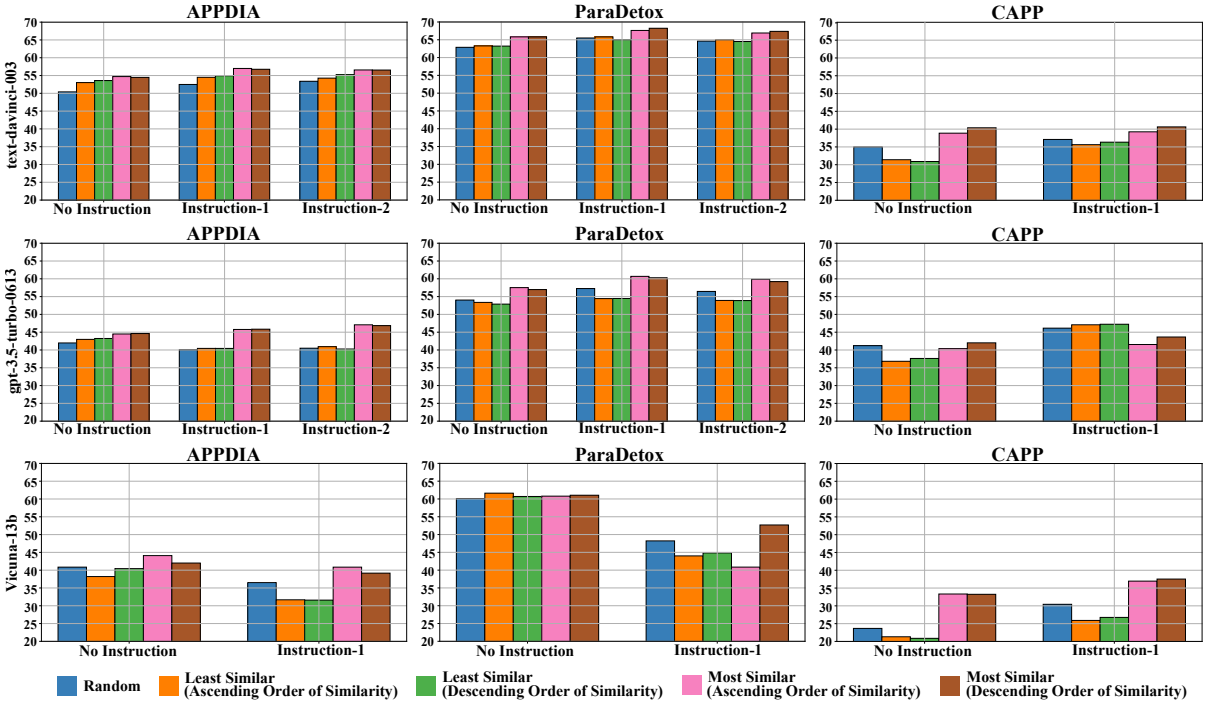

Figure 3: BLEU as a function of order of demonstrations and type of instruction used in the prompt design. Demonstrations that are semantically more similar to the query sample show better performance than less semantically similar and randomly selected samples. Also, prompts that only include demonstrations (*i.e., No Instruction*) show a BLEU score that is comparable to prompts that include instruction and demonstrations.

13b model across all datasets. Further increasing the demos only results in slight improvement, as each additional demo is semantically less similar to the query and thereby less important than the demonstrations selected before (Liu et al., 2021).

We notice in the case of the gpt-3.5-turbo-0613 model on CAPP dataset that BLEU without any demos is better than with 40 demos. It's possible that the main instruction used here was less effective in the ICL paradigm, and that a different instruction could have increased the BLEU score, as seen later in Section 3.3. However, we believe this happens because the Gold-Standard for CAPP was also generated using gpt-3.5-turbo-0613. This hints at the possibility of ICL not necessarily improving paraphrasing performance of LLMs, which in turn were used to generate the dataset. We see similar observations in the following sections as well.

3.2 Selection and Order of Demonstrations

We now discuss the effect of selection and ordering the demos in the prompt on BLEU. Note, in Figures 3 and 4, the number of demonstrations was set to 10 and explore the different ordering mechanisms described in Section 2.2. In Figure 3, we observe that the *Random* strategy sometimes achieves better BLEU than the *Least Similar* strategy. While in most cases the *Most Similar* shows better performance than both *Random* and *Least Similar*. This intuitively makes sense since *Most Similar* represents samples from the training corpus that are most semantically similar to the query (Liu et al., 2021). This enables the LLM to generate a paraphrase that is also similar to the Gold-Standard paraphrase of the query. Next, the order in which the demos are arranged also has an impact on BLEU score. We find that curating the demos in decreasing order of similarity often results in better BLEU than arranging them in increasing order of similarity. We also observe similar trends with other automated evaluation metrics. Note, the above observations do not apply to the gpt-3.5-turbo-0613 model on the CAPP dataset.

While the approaches described in Section 2.2 are simple and effective, they might not bring out the best possible performance. Sophisticated methods to select and order demonstrations have been proposed and have shown better performance in other applications (Ye et al., 2022; Zhang et al., 2022; Lu et al., 2021). However, we find that they offer only marginal improvement, while taking significantly longer times to process each query sample. For example, Table 3 shows the different performance metrics and compute times for (Ye et al.,

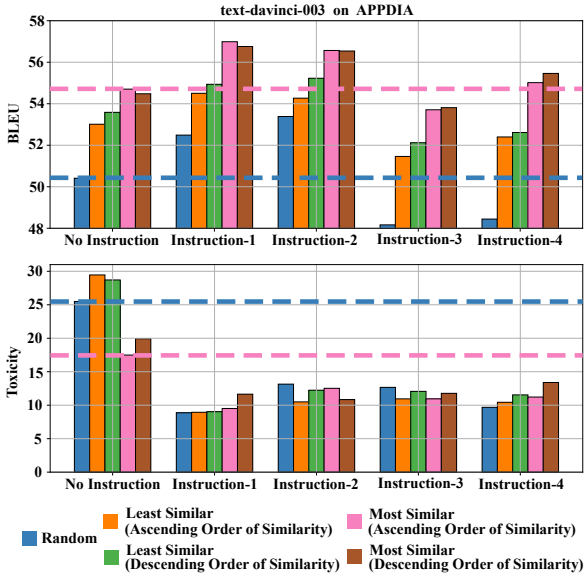

Figure 4: BLEU score and measured toxicity performance with different instructions but with the same set of demos. Instructions can either complement or work against the selected demos and accordingly affect the BLEU score. The *No Instruction* setting shows comparable BLEU to prompts that include both instructions and demos but result in paraphrases with higher toxicity. The dotted reference lines are used to indicate the range in BLEU score under the *No Instruction* setting.

2022)'s MMR method and the proposed ICL-based approach. Note, gpt-3.5-turbo-1106 was used as the generation model for MMR. While MMR might be marginally better quantitatively, it is several orders of magnitude slower than the proposed approach in terms of mean Similarity Compute Time and mean Demo Retrieval Compute Time. Please refer to Appendix C.2 for more details on this topic.

3.3 Significance of Instruction

We now investigate the effect of removing the instruction in the prompt. The left-most set of bars within each bar plot in Figures 3 and 4, which show prompts with *No Instruction*, display BLEU scores that are on par with prompts that include both instructions and demonstrations. This is interesting since it is a common practice to always include the instruction in the prompt even if no demonstrations are provided. Our results suggest that when it is difficult to determine effective instructions for the target paraphrasing task, with ICL one can simply use a few systematically selected demonstrations to get high quality generated paraphrases.

In Figure 4, for text-davinci-003 on APPDIA, we observe that the *No Instruction* setting retains a significant amount of the original content's toxicity, thereby making its usability questionable. Similar observations were made with other models (refer to Appendix C.3, Figure 9). Order of demos also plays an important role in the *No Instruction* setting, with the *Most Similar* showing much lower toxicity than both *Random* and *Least Similar* strategies. For cases that include both instruction and demos, the measured toxicity is less impacted by the order of demos, indicating that the main instruction serves as a toxicity regularizer.

We want to also highlight that creating a good instruction for paraphrasing tasks is non-trivial. Despite using good demos, a bad instruction can negatively impact the quality of the generated paraphrase. For instance, in Figure 3, the Vicuna-13b model shows better BLEU with just the curated demonstrations on the APPDIA and ParaDetox datasets. Similarly, in Figure 4, we see that certain instructions can result in lower BLEU than prompts that have *No Instruction*.

3.4 Comparison with Supervised Approaches

We compare our ICL-based approach to prior state-of-the-art supervised baselines. For APPDIA we use BART, T5, DialoGPT, and PDTB+RST methods as done in (Atwell et al., 2022); for ParaDetox we use BART as done in (Logacheva et al., 2022); and for CAPP we fine-tuned BART-base and T5-base on the training set. We used the default hyperparameters defined in the Transformers Seq2SeqTrainer for fine-tuning on CAPP. The comparison between our ICL-based approaches (including the newer OpenAI models, gpt-3.5-turbo-instruct and gpt-3.5-turbo-1106) and prior baselines is shown in Table 3.

Note, the objective of any paraphraser should be to score high on generation quality and have a low Toxicity in the generated paraphrases. For APPDIA and ParaDetox, the BART and T5 models perform better than the ICL-based approach on the different standard evaluation metrics. However, the paraphrases generated by these baselines seem to retain a significant amount of the original toxicity. To better understand this issue, we use the Toxicity for the *Inoffensive Gold-Standard* in each dataset as a point of reference. Ideally, a paraphraser should generate paraphrases whose average Toxicity is no greater than this reference. We observe that all baseline methods except DialoGPT show a higher Toxicity, while the ICL-based methods exhibit Toxicity that is lower or on par with that of the Gold-Standard. Our approach offers a better trade-off between generation quality and Toxicity.

Dataset	Method	BLEU↑	BERT-F1↑	ROUGE↑	CIDEr↑	Toxicity↓	Quality↑	Similarity Compute Time↓	Demo Retrieval Compute Time↓
APPDIA	*Offensive Test Set*	-	-	-	-	75.60	-	-	-
	Inoffensive Gold-Standard	-	-	-	-	14.37	3.68±0.93		
	BART (Atwell et al., 2022)	65.0	68.1	65.6	4.77	25.91	3.42±1.08		
	T5 (Atwell et al., 2022)	65.3	69.2	66.5	4.75	20.15	-		
	DialoGPT (Atwell et al., 2022)	42.3	46.7	38.0	1.11	14.51	3.52±0.93		
	PDTB+RST (Atwell et al., 2022)	46.2	50.7	42.5	1.54	16.39	-		
	MMR-BERT (10 Demos)	58.5	65.1	59.3	3.90	17.54	-	4.7538	0.0792
	MMR-Embedding (10 Demos)	57.9	63.8	57.7	3.79	14.11	-	0.0025	0.0786
	gpt-3.5-turbo-1106 (10 Demos)	57.1	64.0	57.20	3.72	**14.42**	-	**0.0025**	**0.0005**
	gpt-3.5-turbo-1106 (40 Demos)	61.2	66.4	61.9	4.23	**19.55**	-		
	gpt-3.5-turbo-0613 (10 Demos)	45.8	53.3	41.6	2.12	**7.00**	**4.24±0.91**		
	gpt-3.5-turbo-0613 (40 Demos)	50.4	58.2	47.6	2.67	**10.08**	**4.11±1.00**		
	gpt-3.5-turbo-instruct (10 Demos)	51.9	58.9	50.5	2.81	**15.86**	-		
	gpt-3.5-turbo-instruct (40 Demos)	56.2	62.9	55.8	3.42	**21.37**	-		
	text-davinci-003 (10 Demos)	56.8	63.6	57.6	3.70	**11.64**	**3.98±1.05**		
	text-davinci-003 (40 Demos)	60.9	66.7	62.9	4.29	**12.67**	**3.77±1.08**		
	Vicuna-13b (4 Demos)	38.2	46.8	34.9	1.41	**12.07**	**3.87±1.00**		
	Vicuna-13b (10 Demos)	40.8	48.0	37.6	1.79	**18.44**	**3.91±1.07**		
ParaDetox	*Offensive Test Set*	-	-	-	-	88.64	-	-	-
	Inoffensive Gold-Standard	-	-	-	-	6.56	3.77±0.97		
	BART (Logacheva et al., 2022)	77.3	76.2	69.8	4.94	82.00	2.82±0.75		
	MMR-BERT (10 Demos)	68.6	68.0	58.8	3.67	8.47	-	4.7538	0.5990
	MMR-Embedding (10 Demos)	67.6	67.3	57.7	3.52	8.91	-	0.0025	0.6010
	gpt-3.5-turbo-1106 (10 Demos)	67.6	67.3	57.2	3.45	**8.46**	-	**0.0025**	**0.0048**
	gpt-3.5-turbo-1106 (40 Demos)	70.1	69.3	59.6	3.79	**9.64**	-		
	gpt-3.5-turbo-0613 (10 Demos)	60.3	62.0	50.5	2.72	**5.71**	**3.90±1.01**		
	gpt-3.5-turbo-0613 (40 Demos)	64.3	65.1	54.1	3.08	**6.20**	**3.92±1.02**		
	gpt-3.5-turbo-instuct (10 Demos)	65.2	66.4	55.6	3.17	**9.97**	-		
	gpt-3.5-turbo-instuct (40 Demos)	69.1	68.1	58.9	3.68	**12.3**	-		
	text-davinci-003 (10 Demos)	68.2	67.7	58.9	3.67	**6.50**	**4.34±0.91**		
	text-davinci-003 (40 Demos)	70.1	69.3	60.4	3.95	**6.21**	**4.22±0.96**		
	Vicuna-13b (4 Demos)	49.3	54.1	41.1	1.78	**7.23**	**4.00±0.99**		
	Vicuna-13b (10 Demos)	52.8	56.7	43.7	2.05	**9.98**	**4.53±0.84**		
CAPP	*Offensive Test Set*	-	-	-	-	25.87	-	-	-
	Inoffensive Gold-Standard	-	-	-	-	0.94	4.38±0.83		
	BART	38.5	48.3	36.3	1.86	3.54	3.78±0.87		
	T5	39.4	50.2	37.9	1.92	2.63	3.84±0.87		
	MMR-BERT (10 Demos)	45.5	54.4	41.8	2.20	1.05	-	4.7538	0.3937
	MMR-Embedding (10 Demos)	44.0	52.4	39.9	2.10	1.34	-	0.0025	0.3936
	gpt-3.5-turbo-1106 (10 Demos)	43.9	53.2	40.5	2.17	**1.22**	-	**0.0025**	**0.0031**
	gpt-3.5-turbo-1106 (40 Demos)	45.8	54.8	42.5	2.29	**1.30**	-		
	gpt-3.5-turbo-0613 (10 Demos)	43.7	51.9	39.6	2.00	**0.82**	**4.44±0.81**		
	gpt-3.5-turbo-0613 (40 Demos)	47.1	55.0	43.0	2.33	**0.72**	**4.58±0.76**		
	gpt-3.5-turbo-instruct (10 Demos)	44.9	53.6	41.1	2.18	**1.23**	-		
	gpt-3.5-turbo-instruct (40 Demos)	48.0	56.4	44.5	2.53	**1.49**	-		
	text-davinci-003 (10 Demos)	40.6	49.6	36.1	1.73	**1.04**	**4.03±0.96**		
	text-davinci-003 (40 Demos)	44.5	53.2	40.7	2.10	**1.09**	**4.10±0.94**		
	Vicuna-13b (4 Demos)	35.8	42.4	31.3	1.34	**1.04**	**4.36±0.78**		
	Vicuna-13b (10 Demos)	37.5	35.9	33.6	1.55	**1.02**	**4.21±0.88**		

Table 3: Quantitative, qualitative and mean compute time (in seconds) assessment of different LLMs using the ICL paradigm and comparison against different baseline supervised approaches. Toxicity of the offensive test set and inoffensive ground-truth paraphrases is also provided. Differences in the reported Mean±Std Quality scores between each ICL-based approach and the different baselines is significantly different (*i.e.,* $p-\text{value} < 0.05$).

Figure 5 illustrates the Toxicity measured for the different ICL-based methods by varying the number of demonstrations. The *Most Similar (Descending Order)* strategy was used to select and organize the demos. It also displays the measured Toxicity of the Offensive Test Set, the Gold-Standard and the different baselines. Note that, B1, B2, B3, B4 refer to BART, T5, DialoGPT and PDTB+RST models respectively. For APPDIA and ParaDetox we observe that LLMs without any demonstrations show much lower Toxicity than when any demonstration is used. The reverse, however, is observed for the CAPP dataset. The absence of demos causes LLMs to fallback on their own task definition which results in paraphrases with Toxicity significantly different from that of the Gold-Standard. However, the absence of demos also causes the generated paraphrases to exhibit a lower BLEU score as seen earlier in Figure 2. A balance between the main instruction and demos can ensure generation of paraphrases that reduce offensiveness and score high using different automated metrics.

3.5 Additional Dialogue Context Helps

We show preliminary results of using the prior utterances leading up to the rude utterance as additional context in the our ICL-based method. Similar to the example in Section 2.1, we prepend the context for both the demo and the query. Figure 6 shows BLEU score as we add/remove context and vary the number of demonstrations. We clearly see performance improvement by incorporating dialogue context us-

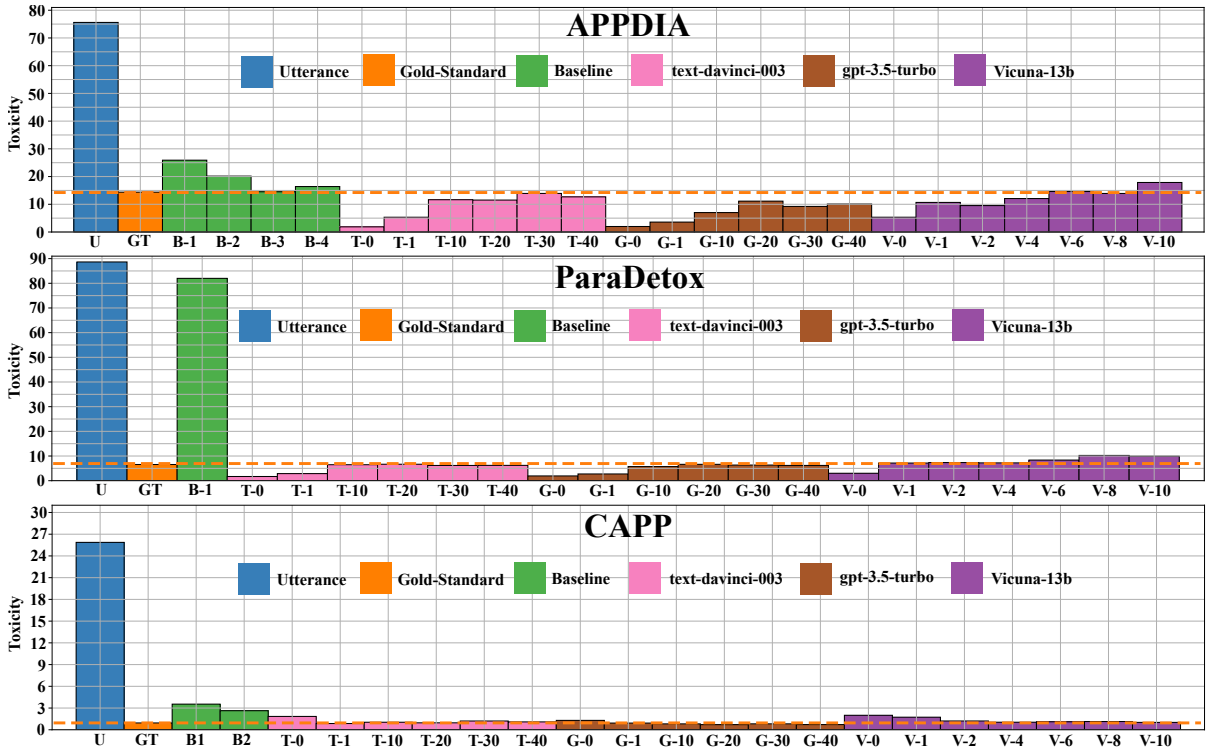

Figure 5: Average Toxicity measured using the *Detoxify* (Hanu and Unitary team, 2020). The orange dotted line serves as a reference for the Gold-Standard's Toxicity. U, GT, B-#, T-#, G-#, V-# along the x-axis refer to Utterance, Gold-Standard, Baseline methods, text-davinci-003, gpt-3.5-turbo, Vicuna-13b respectively. # in T-#, G-#, V-# indicate number of demonstrations used. Note, T-0, G-0, V-0 only contain an instruction in the prompt.

ing the text-davinci-003 and gpt-3.5-turbo-0613 models. We were unable to create a prompt for Vicuna-13b that successfully uses additional context in the ICL framework. We will develop such prompts for Vicuna-13b in future work.

3.6 Robustness to Reduced Training Data

Here we study the impact of available training data on the performance of our best performing strategy i.e. *Most Similar (Descending Order)*. We observe only a minimal fall in BLEU up to 10% of the training data as shown for text-davinci-003 in Figure 7 (refer to Appendix C.4, Figure 10 for other models). Further reducing training data results in noticeable drop in BLEU. We also find that reducing training dataset below 10% results in BLEU score that is similar to *Random* demo selection and arrangement strategy but with access to 100% of the training data. That result shows that our ICL-based method can work with limited training data and thus can be adapted quickly to novel settings.

3.7 Manual Qualitative Assessment

We also perform quality assessment of the Gold-standard and generated paraphrases using a human annotator. We select a subset of 150, 200, and 200 samples from the test-set of APPDIA, ParaDetox and CAPP respectively. For ParaDetox and CAPP, we use all the supervised baseline methods listed in Table 3. For APPDIA we only use BART and DialoGPT models for comparison. Our in-house annotator (mentioned earlier in Section 2.3) used the scoring guidelines described in Tables 1 and 4. Information about the type of model used to generate each paraphrase was not made available to the annotator. Table 3 shows that the three ICL-based LLM models received a higher average score that is significantly different (*i.e.,* p-value < 0.05) than the corresponding baseline methods. We also note that Vicuna-13b's qualitative score was comparable to and in some cases better than the OpenAI models, despite having scored lower on metrics that are traditionally used to measure generation quality. This shows that open-source LLM paraphrasers are comparable to closed-source LLMs as per human assessment and might be a viable alternative. Refer to Appendix C.5 for additional analysis between manual evaluation score and toxicity metric.

4 Related Work

Our paper explores the potential use of LLMs with ICL for paraphrasing systems. There has been sig-

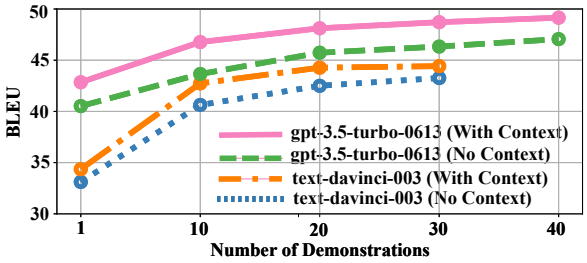

Figure 6: Comparison of BLEU performance on CAPP between including and excluding prior context in the form of prior dialogue utterances.

Score	Description
5	Perfect meaning-preserving inoffensive paraphrase.
4	Paraphrase that is inoffensive but somewhat distinct in meaning.
3	Meaning-preserving paraphrase that could be less offensive.
2	Paraphrase that is very different in meaning and somewhat less offensive than the original.
1	Paraphrase that is very different in meaning and not less offensive than the original.

Table 4: Description of the scoring guidelines used in Section 3.7 for evaluating the generated paraphrases for the APPDIA (Atwell et al., 2022) and ParaDetox (Logacheva et al., 2022) datasets.

nificant interest in better understanding the capabilities of ICL, but for other applications (Min et al., 2021a; Zhao et al., 2021; Razeghi et al., 2022; Xie et al., 2021; Lampinen et al., 2022; Mishra et al., 2021; Chen et al., 2021; Min et al., 2021b; Chen et al., 2023). For instance, (Lu et al., 2021) showed that order of demos has a significant impact on model performance. (Liu et al., 2021) showed that retrieving demonstrations that are semantically similar to the query can be a more effective approach to control the variability in performance. (Rubin et al., 2021) learned an encoding scheme to retrieve better demos for ICL. Other works also explored the influence of number of demos in different settings (Garg et al., 2022; Min et al., 2022; Wei et al., 2023). (Zhou et al., 2022) evaluated the importance of each part in the prompt has towards the final performance. In this paper we study the impact of various components on the final performance, while ensuring that the toxicity of the outputs is within tolerable levels. This enables us to propose a few-shot solution to offensive content paraphrasing. Most prior works (Atwell et al., 2022; Logacheva et al., 2022) have modeled paraphrasing as a sequence-to-sequence problem and trained models such as T5, BART on human annotated data. Despite good generation results, these models tend towards higher toxicity and are difficult to adapt to new applications without collecting more data. Our solution addresses those challenges successfully, with only a fraction of the original training set.

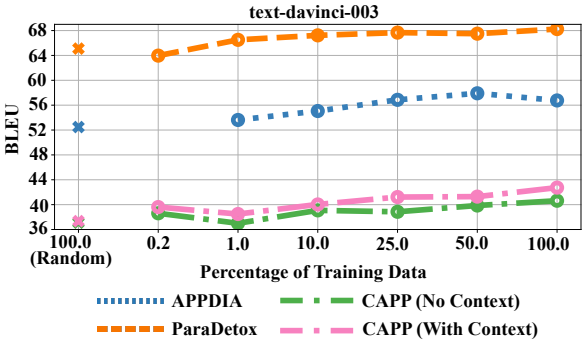

Figure 7: BLEU for the *"Most Similar (Descending Order)"* approach as a function of percentage of training data available, and comparison with Randomly selected demonstrations using 100% of the training data.

5 Conclusion

In this paper, we focus on developing usable offensive content paraphrasing systems by leveraging generalization capabilities of LLMs and quickly adapting them to new tasks using ICL. A paraphraser should generate qualitatively good paraphrases that preserve the original content's meaning, while also minimizing toxicity. Focusing only on one of these aspects compromises overall usability. Compared to supervised approaches that require lot of training data and often produce undesired yet coherent paraphrases, our ICL-based framework is generally comparable on various evaluation metrics like BLEU, but is qualitatively better and helps significantly reduce toxicity in the generated paraphrases. Through systematic experiments we tested the capabilities and limitations of ICL-based offensive paraphrasers. Other key highlights of using our ICL framework include: (1) Selection and arrangement of demos significantly impacts quality of paraphrases; (2) Measured toxicity is lowest when only the instruction is used and highest when only demos are used. Combining both instruction and demos helps ensure quality and usability of generated paraphrases; (3) Robust to limited data, *i.e.,* with just 10% training data we only see a slight decrease in overall performance, thereby enabling us to easily scale and deploy.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001122C0032. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views or policies of DARPA, the Department of Defense or the U.S. Government.

Limitations

Here, we list the limitations identified in this paper:

1. We found that ICL fails on datasets that were prepared using the same LLMs used in the ICL framework. Since we used the gpt-3.5-turbo-0613 model to create polite paraphrases for our CAPP dataset, we were unable to see the same observations in our results when using other models and datasets. This could have been avoided by creating manually annotated polite paraphrases. However, manual annotation is a laborious process and isn't scalable. Hence, a manual qualitative assessment was done on a small subset of the final CAPP dataset to ensure usability of the generated paraphrases.

2. Prompt engineering for the Vicuna-13b model with the ICL framework is nontrivial. We found it difficult to create main instructions in the prompt that result in the Vicuna-13b model to behave in a desired way. Also, unlike the two OpenAI models, the number of demonstrations that can be effectively passed into Vicuna-13b is quite limited. In some cases we were able to concatenate more than 10 demos to the prompt but it often resulted in generating incomprehensible or empty outputs.

3. The *No Instruction* prompt explored in the paper resulted in paraphrases that are comparable to prompts that include both instruction and demos, on several automated evaluation metrics. However, we notice that the *No Instruction* setting also retains a significant amount of toxicity from the original content. We propose that in situations where it is difficult to decide on a good main instruction, one could simply use a few carefully curated and ordered demos like the "Most Similar (Descending Order)" approach to generate paraphrases and check if it is within the desired toxicity levels.

4. Our experimental results indicate that there is no single prompt that works in all situations. One must carefully balance the main instruction and the set of demos from the training corpus to get desired paraphrase outputs.

5. We showed preliminary results showcasing the benefit of incorporating additional context in the form of prior two utterances in the ICL framework. We believe there can be better ways to incorporate this contextual information and further improve performance of LLMs.

6. The closed-source OpenAI models are more powerful, faster and expensive to use. Despite open-source models like Vicuna-13b coming close to OpenAI models on other tasks, they still have a long way to go for offensive content paraphrasing.

Ethics Statement

We have to take great care with our collection of offensive content to protect privacy. We have to ensure judicious use of the collected data to protect the vulnerable against such speech. We recognize that our models cannot entirely eliminate offensive content from a given text. Additionally, we acknowledge that utilizing pretrained models may introduce biases in specific situations, as studies have revealed that pretrained models can be influenced by biases present in the data used for their initial training. We have to continue research on making sure that the LLMs do not hallucinate and end up injecting toxicity since we don't know what they have been trained on. There is a danger of this kind of technology being used in reverse, *i.e.*, take harmless content and paraphrase to inject toxicity. We realize that ethics is an ongoing challenge. We are engaged with the Fairness, Accountability and Transparency community and are learning to address key ethics issues on an ongoing basis.

References

Katherine Atwell, Sabit Hassan, and Malihe Alikhani. 2022. Appdia: A discourse-aware transformer-based style transfer model for offensive social media conversations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6063–6074.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Pete Burnap and Matthew L Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet*, 7(2):223–242.

Derek Chen, Celine Lee, Yunan Lu, Domenic Rosati, and Zhou Yu. 2023. Mixture of soft prompts for controllable data generation. *arXiv preprint arXiv:2303.01580*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2021. Meta-learning via language model in-context tuning. *arXiv preprint arXiv:2110.07814*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. *arXiv preprint arXiv:1306.6078*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Bryan Dosono and Bryan Semaan. 2019. Moderation practices as emotional labor in sustaining online communities: The case of aapi identity work on reddit. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13.

Antigoni Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2019. A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM conference on web science*, pages 105–114.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598.

Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

Sabit Hassan, Katherine J Atwell, and Malihe Alikhani. 2022. Studying the effect of moderator biases on the diversity of online discussions: A computational cross-linguistic study. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Shagun Jhaver, Darren Scott Appling, Eric Gilbert, and Amy Bruckman. 2019. Did you suspect the post would be removed? understanding user reactions to content removals on reddit. *Proceedings of the ACM on human-computer interaction*, 3(CSCW):1–33.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pages 1621–1622.

Andrew K Lampinen, Ishita Dasgupta, Stephanie CY Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L McClelland, Jane X Wang, and Felix Hill. 2022. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021a. Noisy channel language model prompting for few-shot text classification. *arXiv preprint arXiv:2108.04106*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021b. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Jeremy Waldron. 2012. *The harm in hate speech*. Harvard University Press.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. 2014. Cursing in english on twitter. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 415–425.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

John Wieting and Kevin Gimpel. 2017. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*.

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1980–1984.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Meng Ye, Karan Sikka, Katherine Atwell, Sabit Hassan, Ajay Divakaran, and Malihe Alikhani. 2023. Multilingual content moderation: A case study on reddit. *arXiv preprint arXiv:2302.09618*.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2022. Complementary explanations for effective in-context learning. *arXiv preprint arXiv:2211.13892*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. *arXiv preprint arXiv:2211.04486*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

A Context-Aware Polite Paraphrase (CAPP) Dataset

This section provides additional details about the training protocol used to develop the rudeness detector described in Section 2.3. The initial rudeness detection model was trained on a randomly-selected subset of the OpenSubtitles corpus (Lison and Tiedemann, 2016) that was manually annotated by an in-house human annotator, as well as the Stanford Politeness corpus (Danescu-Niculescu-Mizil et al., 2013). Between these two data sources, this model was trained on 10997 gold-quality samples. This intermediate model achieved an overall accuracy score of 59.40 across the three classes (rude, polite, and neutral). The accuracy for specifically rude utterances in the test data was 71.84 – higher than that of other classes. The final rudeness detection model, trained on a combination of the aforementioned gold data, and a larger subset (250379 samples) of the OpenSubtitles corpus annotated using the intermediate model, achieved an overall accuracy of 57.20 across the three classes, with an accuracy score of 71.84 on rude samples – still higher than the accuracy on other classes. The disparity in rudeness detector performance between classes was deemed acceptable due to its high performance on the rude class. Accurate identification of rude utterances was prioritized in the interest of generating a sufficient number and variety of rude utterances for paraphraser training – with an acceptance of the potential for noise in the form of some non-rude utterances in the training data.

B Instructions Used in Experiments

The exact prompt instruction used for each experiment in Section 3 is shown below. Except Subsection 3.2 and 3.3, we use only **Instruction-1** for the following Subsections 3.1, 3.4, 3.5, 3.6 and 3.7.

1. Instruction prompts used for text-davinci-003 and gpt-3.5-turbo models on the APPDIA dataset:
   (a) **Instruction-1:** Instruction: Rewrite the following sentence to be inoffensive
   (b) **Instruction-2:** Instruction: Rewrite the following sentence to be inoffensive and ensure that the paraphrased sentence is similar to the original sentence
   (c) **Instruction-3:** Instruction: Generate a paraphrased sentence that expresses a more balanced viewpoint
   (d) **Instruction-4:** Instruction: Rewrite the following sentence with a more objective tone
2. Instruction prompts used for text-davinci-003 and gpt-3.5-turbo models on the ParaDetox dataset:
   (a) **Instruction-1:** Instruction: Rewrite the following sentence to be inoffensive
   (b) **Instruction-2:** Instruction: Rewrite the following sentence to be not toxic
3. Instruction prompts used for text-davinci-003, gpt-3.5-turbo and Vicuna-13b models on the CAPP dataset:
   (a) **Instruction-1:** The following sentence can contain rude text. Please provide one paraphrased sentence that is semantically similar to it and is polite.
4. Instruction prompts used for Vicuna-13b model on APPDIA and ParaDetox datasets:
   (a) **Instruction-1:** The following sentence contains offensive text. Please provide one paraphrased sentence that is semantically similar to it and is inoffensive.
5. While incorporating additional context in the form of previous two turns in the dialogue, the instruction prompt used for text-davinci-003 and gpt-3.5-turbo models on the CAPP dataset:
   (a) **Instruction-1:** Paraphrase only the below Sentence to be polite and semantically similar to the Sentence. Use the context as as reference but do not include any part of it in the final paraphrase.

C Experiments

C.1 Number of Demonstrations

Figure 8 provides additional details, supporting the information described in Section 3.1.

C.2 Selection and Order of Demonstrations

This section provides additional details, supporting the information described in Section 3.2. While the original paper (Ye et al., 2022) suggests using BERT-Score for similarity measurement, we also explore the cosine similarity measurement for the normalized embeddings extracted using the sentence transformer models. We refer to the original

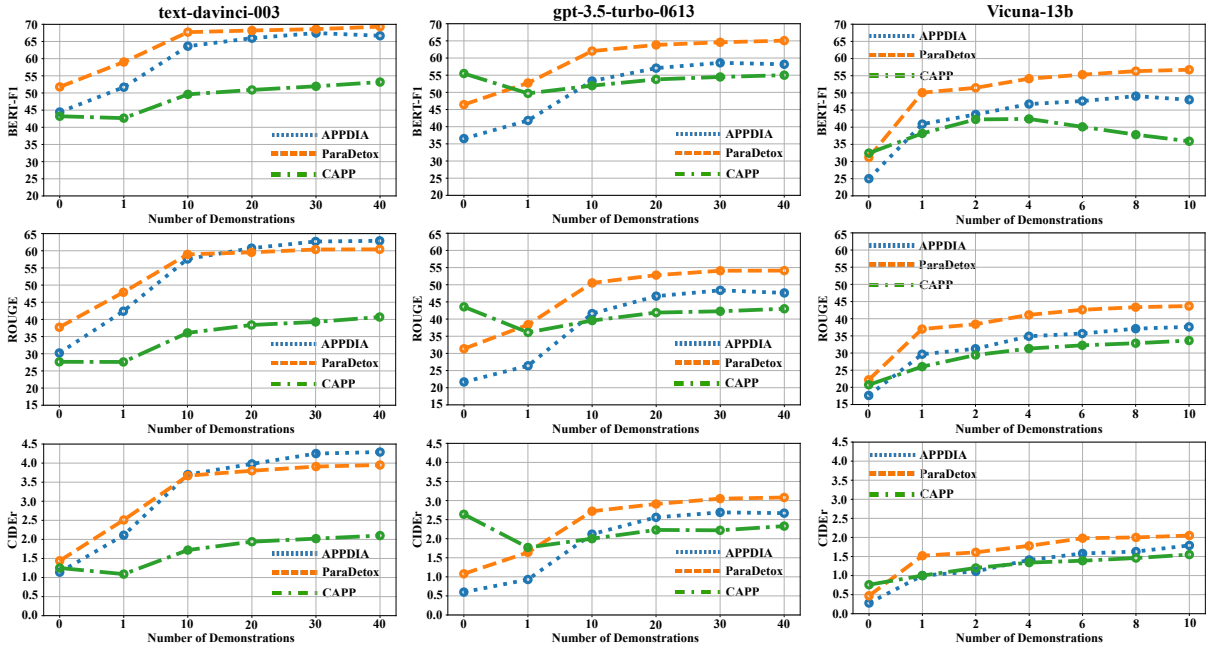

Figure 8: Performance using different evaluation metrics as a function of number of demonstrations used in the prompt. Noticeable improvement in score performance is observed in the beginning, with performance saturating after a certain number of demos.

Dataset	Method	Similarity Compute Time↓	Demo Retrieval Compute Time↓
APPDIA	MMR-BERT (10 Demos)	4.7538	0.0792
	MMR-BERT (40 Demos)	4.7538	0.5668
	MMR-Embedding (10 Demos)	0.0025	0.0786
	MMR-Embedding (40 Demos)	0.0025	0.5569
	gpt-3.5-turbo-1106 (10 Demos)	**0.0025**	**0.0005**
	gpt-3.5-turbo-1106 (40 Demos)	**0.0025**	**0.0005**
ParaDetox	MMR-BERT (10 Demos)	4.7538	0.5990
	MMR-BERT (40 Demos)	4.7538	4.39
	MMR-Embedding (10 Demos)	0.0025	0.6010
	MMR-Embedding (40 Demos)	0.0025	4.17
	gpt-3.5-turbo-1106 (10 Demos)	**0.0025**	**0.0049**
	gpt-3.5-turbo-1106 (40 Demos)	**0.0025**	**0.0047**
CAPP	MMR-BERT (10 Demos)	4.7538	0.3937
	MMR-BERT (40 Demos)	4.7538	2.97
	MMR-Embedding (10 Demos)	0.0025	0.3936
	MMR-Embedding (40 Demos)	0.0025	3.06
	gpt-3.5-turbo-1106 (10 Demos)	**0.0025**	**0.0031**
	gpt-3.5-turbo-1106 (40 Demos)	**0.0025**	**0.0031**

Table 5: Comparison of the proposed demonstration selection and ordering approach to MMR (Ye et al., 2022). Here, the proposed approach refers to the *Most Similar (Descending Order)* approach outlined in Section 3.2. While MMR provides marginal performance gains on all three datasets as shown in Table 3, it is several orders of magnitude slower than the proposed approach. Here, the mean *Similarity Compute Time* measures the average time taken to perform similarity measurement between a query test sample and all the available reference training samples; The mean *Demo Retrieval Compute Time* measures the average time taken to select n demonstrations from the available training set based on the similarity measurements done previously.

implementation as MMR-BERT and name the new variant as MMR-Embedding. The parameter λ was set to 0.5 for both MMR approaches. Tables 3 and 5 compare the performance differences between the different MMR approaches and the proposed best performing ICL-based approach, *i.e.*, the *Most Similar (Descending Order)* method. Comparison is done using various quantitative evaluation metrics and compute time metrics. Note, for the proposed approach, the compute times reported correspond

to the *Most Similar (Descending Order)* method, however, these would also be the same for the other demonstration selection and ordering approaches described in Section 3.2. The different compute time metrics explored are defined as follows – The mean *Similarity Compute Time* measures the average time taken to perform similarity measurement between a query test sample and all the available reference training samples; The mean *Demo Retrieval Compute Time* measures the average time taken to select n demonstrations from the available training set based on the similarity measurements done previously. Here, n is defined beside each method's name within parenthesis. The total time taken to process each query test sample would approximately be equal to the sum of the above two compute times.

Table 3 shows that the MMR methods offer marginal improvement with respect to the different quantitative metrics but at the expense of significant time delays. Note, the Similarity Compute Time for MMR-Embedding is the same as our proposed ICL-based approach since both employ the cosine similarity metric. Quantitative metrics for the two MMR approaches is only reported for 10 demonstrations in Table 3 as we observed even greater delays with respect to Demo Retrieval Compute Time when trying to retrieve 40 demonstrations. Table 5 reports the two Compute Times for the different MMR approaches and proposed ICL method, for both 10 and 40 demonstrations. Note, gpt-3.5-turbo-1106 was used as the generation model for MMR. For the proposed ICL-based approach we observe negligible differences in compute times when we want to retrieve either 10 or 40 demonstrations. However, the MMR-BERT approach becomes 10 times slower when trying to retrieve 40 demonstrations instead of 10.

While there are several demonstration selection and ordering approaches (Ye et al., 2022; Zhang et al., 2022; Lu et al., 2021) that can help push the performance ceiling, one must also make sure if these approaches can be easily implemented and scaled up in a real-time, real-world application setting. For example, (Zhang et al., 2022) propose a reinforcement learning algorithm to identify generalizable policies to select demonstrations but find that this approach offers diminishing returns on larger, more sophisticated LLMs. The approach described by (Lu et al., 2021) to overcome few-shot prompt order sensitivity is better suited for multi-class classification tasks, since the entropy-based statistics framework discussed to identify performant prompts is not directly applicable to text generation tasks like paraphrasing.

C.3 Significance of Instruction

Figure 9 provides additional details, supporting the information described in Section 3.3.

C.4 Robustness to Reduced Training Data

Figure 10 provides additional details, supporting the information described in Section 3.6.

C.5 Correlation between Manual Evaluation Score and Automated Toxicity Metric

The manual quality evaluation score considers not just Toxicity minimization in the generated paraphrase but also considers meaning preservation. Directly comparing it to the automated toxicity score is not possible since there is a semantic mismatch between the two metrics. Instead, we first compute the difference in the toxicity measured between the (offensive) utterance and the (inoffensive) paraphrase. Next, we compute the Pearson correlation between this difference in toxicity score to the manual quality evaluation score. This difference captures the comparisons made by the annotator while coming up with the manual scoring as shown in Tables 1 and 4. Therefore, if there is correlation between this automatically computed difference and the manual scores, then convergent validity is assured.

Tables 6, 7, 8 show the computed Pearson correlation coefficient between the manual quality evaluation score and automated toxicity score under two different settings. Type-1 represents the Pearson correlation coefficient between the manual quality score and automated toxicity score of the paraphrased output; and Type-2 represents the Pearson correlation coefficient between the manual quality score and difference in measured toxicity between original utterance and paraphrased output. Note, the dynamic range of the toxicity captured in the CAPP dataset is low because it mostly consists of rude speech that contains little or no foul language.

Dataset	Method	Type-1	Type-2
APPDIA	Gold Standard	-0.28	0.07
	BART	-0.41	0.36
	DialoGPT	-0.40	0.28
	text-davinci-003 (10 demos)	-0.45	0.25
	text-davinci-003 (40 demos)	-0.43	0.27
	gpt-3.5-turbo-0613 (10 demos)	-0.39	0.26
	gpt-3.5-turbo-0613 (40 demos)	-0.35	0.21
	Vicuna-13b (4 demos)	-0.42	0.24
	Vicuna-13b (10 demos)	-0.41	0.17

Table 6: Computed Pearson correlation coefficient on the APPDIA dataset between manual quality score and automated toxicity score of paraphrased output, denoted as Type-1; manual quality score and difference in measured toxicity between original utterance and paraphrased output, denoted as Type-2.

Dataset	Method	Type-1	Type-2
ParaDetox	Gold Standard	-0.15	0.19
	BART	-0.60	0.56
	text-davinci-003 (10 demos)	-0.22	0.20
	text-davinci-003 (40 demos)	-0.22	0.28
	gpt-3.5-turbo-0613 (10 demos)	-0.24	0.25
	gpt-3.5-turbo-0613 (40 demos)	-0.26	0.24
	Vicuna-13b (4 demos)	-0.24	0.26
	Vicuna-13b (10 demos)	-0.49	0.41

Table 7: Computed Pearson correlation coefficient on the ParaDetox dataset between manual quality score and automated toxicity score of paraphrased output, denoted as Type-1; manual quality score and difference in measured toxicity between original utterance and paraphrased output, denoted as Type-2.

Dataset	Method	Type-1	Type-2
CAPP	Gold Standard	-0.139	-0.049
	BART	-0.189	-0.094
	T5	-0.188	-0.005
	text-davinci-003 (10 demos)	-0.155	0.022
	text-davinci-003 (40 demos)	-0.218	-0.016
	gpt-3.5-turbo-0613 (10 demos)	-0.197	-0.077
	gpt-3.5-turbo-0613 (40 demos)	-0.265	-0.129
	Vicuna-13b (4 demos)	-0.096	-0.188
	Vicuna-13b (10 demos)	-0.176	-0.058

Table 8: Computed Pearson correlation coefficient on the CAPP dataset between manual quality score and automated toxicity score of paraphrased output, denoted as Type-1; manual quality score and difference in measured toxicity between original utterance and paraphrased output, denoted as Type-2.

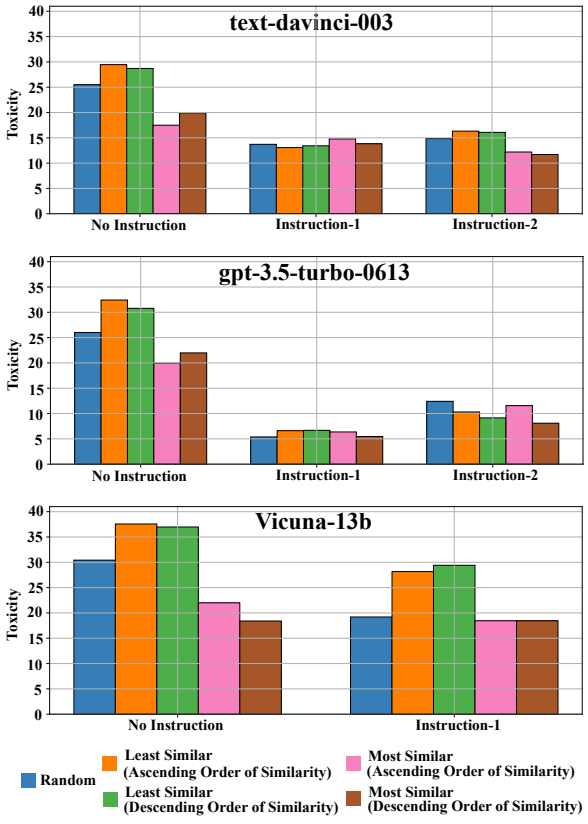

Figure 9: Measured toxicity performance of different models on the APPDIA dataset, with different instructions but with the same set of demos. No instruction setting results in paraphrases with higher toxicity.

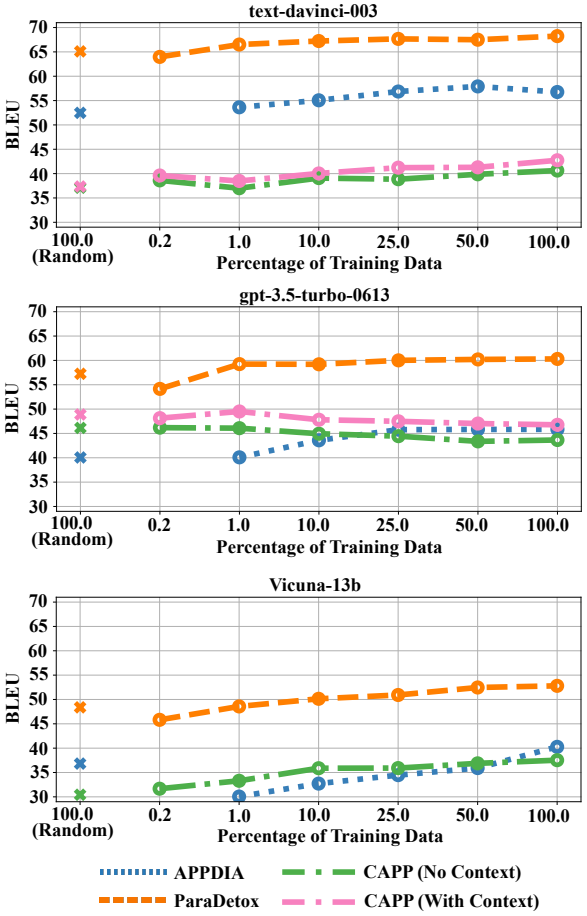

Figure 10: BLEU for the *"Most Similar (Descending Order)"* approach as a function of percentage of training data available and comparison to Random demo selection with access to 100% of the training data.