Assessing the Robustness of Large Language Models At Zero-shot Abstractive Summarization Through the Lens of Relevance Paraphrasing

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have achieved state-of-the-art performance at generating zero-shot summaries from given input articles. However, little is known about the robustness of LLMs at the specific task of zero-shot abstractive summarization. To bridge this gap, we propose *relevance paraphrasing*, a simple strategy that can be used to measure the robustness of LLMs as summarizers. The relevance paraphrasing approach identifies the most *relevant* sentences that contribute to generating an ideal summary, and then *paraphrases* these inputs to obtain a minimally perturbed dataset. Then, by evaluating and comparing model performance for zero-shot summaries generated on both the original and perturbed datasets, we can assess LLM summarization robustness. We conduct extensive experiments with relevance paraphrasing on 4 diverse datasets, as well as 4 LLMs of different sizes (GPT-3.5$_{\text{Turbo}}$, Llama-2$_{\text{13B}}$, Mistral$_{\text{7B}}$, and Dolly-v2$_{\text{7B}}$). Our results indicate that LLMs are not very robust summarizers, as performance drops consistently for the minimally perturbed articles, necessitating further improvements.

1 Introduction

Large Language Models (LLMs) have achieved tremendous success at a number of natural language tasks such as question answering (Robinson and Wingate, 2022), computer program generation (Vaithilingam et al., 2022), and text summarization (Zhang et al., 2023), among others. In particular, modern LLMs have made remarkable progress in generating *abstractive* summaries from input articles that are comparable to summaries written by humans (Zhang et al., 2023). However, while *best-case* performance of LLMs at zero-shot summarization is clearly superlative to other neural models, relatively little is known about the *robustness* of their performance at this task.

Previous work on LLM robustness has primarily investigated *adversarial robustness* by evaluating them on adversarial prompts meant to induce unsafe behavior (Zhu et al., 2023a; Wang et al., 2021). Similarly, a number of adversarial attacks have been proposed for LLMs for various threat models (Jones et al., 2023; Zou et al., 2023) based on manual engineering or prompt optimization. However, our goal in this work differs conceptually from an adversarial attack– we aim to measure *general* robustness performance using a novel paraphrasing strategy which does not have knowledge of the target LLM being used. In contrast, adversarial attacks seek to induce *worst-case* LLM performance by crafting adversarial inputs specific to the model. Note that these attacks target the instruction following capabilities of LLMs, and summarization-specific attacks have not yet been proposed.

Other works (Ye et al., 2023b; Ko et al., 2023) have raised concerns of variability in existing LLM benchmarks and an overall lack of performance credibility (for instance, due to known issues of test set leakage into training data) to measure robustness by proposing novel *evaluation methods*. There are also a number of position papers (Štefánik, 2022) and surveys (Chang et al., 2023) on robustness in LLMs, but none of these have explored the robustness of LLM performance at the specific task of *zero-shot abstractive summarization*.

In this work, we aim to bridge this gap by proposing a novel method for analyzing the robustness of LLM summarization. For learning tasks, *robustness* has generally been defined (Carlini and Wagner, 2017) as the *change in the magnitude of model performance upon minimally perturbing the input space*. Based on this definition, we formulate and seek to answer the following research question in this work: *how does LLM zero-shot abstractive summarization performance vary with minimal perturbations of the input articles to be summarized?*

To make progress towards this goal of quantitatively assessing LLM robustness at summarization, we propose a novel strategy named *relevance para-*

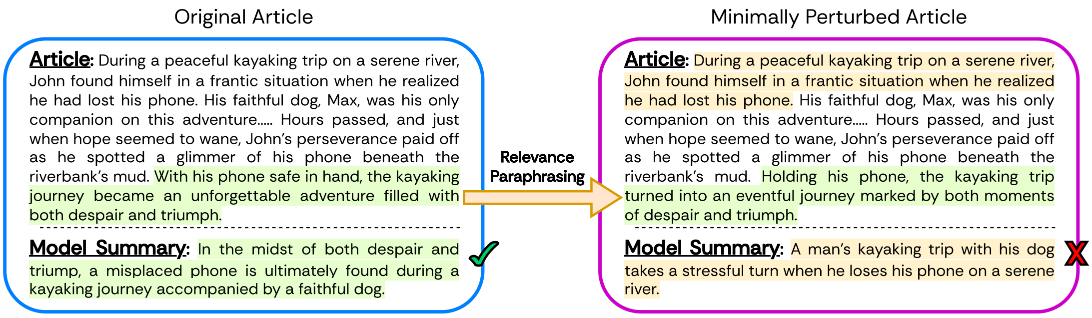

Figure 1: An example showcasing *relevance paraphrasing*. When sentences *relevant* to generating the summary are *paraphrased* to create a minimally perturbed article, we find that zero-shot summarizaton performance drops as the model uses other sentences instead to craft the summary, leading to a loss of salient information.

phrasing for minimally perturbing the input space of articles. Relevance paraphrasing involves identifying which *relevant* sentences from the input article contribute most to generating an ideal gold summary. Then these sentences are *paraphrased* in the article so that they retain semantic meaning to the original version but are phrased differently. This gives us a minimally perturbed version of the input set of articles as only a few sentences are paraphrased. Note that paraphrasing is a simple operation that retains close similarity to the original set of articles so if the LLM is a robust summarizer, its performance should not change much for the perturbed input articles. Thus, by measuring the change in performance on both the original and perturbed set of input articles, we can assess LLM zero-shot summarization robustness. An example of *relevance paraphrasing* is shown in Figure 1.

More importantly, through our analysis of LLM summarization robustness, we wish to draw attention to the need for more work on task-specific robustness analysis of LLMs. As shown in our results in subsequent sections, LLMs tend to exhibit lower performance across a number of different evaluation metrics (such as ROUGE (Lin, 2004) and BertScore (Zhang et al., 2019)) for the perturbed input articles obtained using relevance paraphrasing. We find that post relevance paraphrasing, LLMs select entirely different input article sentences to craft the output summary, losing salient information in the process. This trend is consistently observed across LLMs of different sizes and model parameters[1] as well as multiple datasets. Our results hence indicate that LLMs are not robust summarizers, and necessitate further improvements to ensure more consistent zero-shot summarization performance.

[1]We study GPT-3.5$_{\text{Turbo}}$ (Ye et al., 2023a), Llama-2$_{\text{13B}}$ (Touvron et al., 2023), Dolly-v2$_{\text{7B}}$ (Conover et al., 2023), and Mistral$_{\text{7B}}$ (Jiang et al., 2023) in experiments.

2 Related Works

LLM robustness has largely been studied in the context of adversarial attacks, where a malicious adversary seeks to execute unsafe model behavior by *automatedly* (Zou et al., 2023; Wang et al., 2023; Zhu et al., 2023b) or *manually* optimizing (Wei et al., 2023; Perez and Ribeiro, 2022; Rao et al., 2023) input prompts. Complementary to these efforts, benchmarks have also been proposed to evaluate adversarial robustness of LLMs (Zhu et al., 2023a; Wang et al., 2021). It is important to note that our work contrasts with research on adversarial robustness of LLMs both conceptually and in terms of motivation. Instead of generating worst-case model specific adversarial prompts, we employ model agnostic relevance paraphrasing that minimally perturbs the input articles to characterize *general and natural* robustness of LLMs at the zero-shot summarization task.

Other work on LLM robustness has proposed evaluation methodologies and workflows to assess model performance at general instruction following (Sun et al., 2023) and tasks other than summarization, such as program synthesis (Shirafuji et al., 2023), sentence classification (Ko et al., 2023), and reasoning problems (Ye et al., 2023b). To the best of our knowledge, while a number of works have studied the summarization capabilities of LLMs (Tam et al., 2023; Zhang et al., 2023; Shen et al., 2023), none of these have analyzed the robustness of LLMs at the summarization task, which we seek to assess through our work.

3 Measuring Robustness Via Relevance Paraphrasing

3.1 Zero-Shot Summarization

A zero-shot abstractive summarization model $\mathcal{M}$ takes as input a dataset tuple $T = (X, S^G)$ where

X is a set of articles and S^G are their corresponding *gold standard* summaries, written by human experts. Each article $x \in X$ and gold summary $g \in S^G$ have a variable number of sentences. The model $\mathcal{M}$ then takes in as input the set of articles in the set X and outputs a set of summaries, i.e., $\mathcal{M}(X) = S^{\mathcal{M}}$ where $S^{\mathcal{M}}$ is the set of model generated summaries. Traditionally, the model is evaluated by comparing the generated summaries ($S^{\mathcal{M}}$) with the gold summaries (S^G) using evaluation metrics such as ROUGE (Lin, 2004) and BertScore (Zhang et al., 2019).

3.2 Relevance Paraphrasing

Let an article be denoted as $x \in X$ and its corresponding gold summary is $s \in S^G$. Similar to previous work in abstractive summarization (Kim et al., 2019; Zhao et al., 2022), we assume a proxy mapping function ψ that takes in a (gold) summary sentence $s_i \in s$ and returns a sentence $x_j \in x$ in the article that contributed most to that summary sentence. Any similarity function can be employed as a useful approximation for such a function ψ but in this paper we utilize TF-IDF vector similarities due to computational efficiency and overall accuracy. Also let us assume that we have a paraphrasing model θ that takes in as input a sentence and returns a paraphrased version which retains semantic similarity but is phrased differently. Such a model θ could be a simple strategy such as *active-to-passive*, *formal-to-casual*, or a neural model such as an LLM being used for paraphrasing. In this paper, we use Llama-2_{13B} for this purpose.

The *relevance paraphrasing* process is presented as Algorithm 1. Here, we wish to uncover how robust LLMs are at the task of zero-shot abstractive summarization. In particular, the process works as follows: we first obtain the gold summary for each input article $x \in X$ as $s \in S^G$. Next, we use ψ to obtain a set of article sentences corresponding to each summary sentence in s. Analytically, using ψ for each article-summary pair (x, s), let us maintain a set of indices $I_x = \{j | x_j = \psi(s_i), \forall s_i \in s\}$ which is essentially a set of all the article sentence indices that contributed most to the gold summary.

Now, our goal is to paraphrase each of these *relevant* sentences for article x (that are important for its summary) using the paraphrasing model. We then replace those sentences in the article with their paraphrased versions. That is, for each of these article sentences $x_i, \forall i \in I_x$ we will now obtain a paraphrased version x'_i using the paraphrasing model θ and replace each x_i with paraphrased x'_i to obtain a paraphrased version of the article x'. We then repeat this process to obtain the entire set of paraphrased articles as X'. Now using the difference in obtained model performance we can assess the summarization robustness of LLMs. For instance, if a given evaluation metric $\mathcal{E}$ (such as BertScore) averaged over all test set summaries worsens (e.g. $\mathcal{E}(S^G, \mathcal{M}(X)) > \mathcal{E}(S^G, \mathcal{M}(X'))$) for the paraphrased set of articles compared to the original versions, we can conclude that the LLM performance is not robust.

Algorithm 1 : Relevance Paraphrasing

Input: LLM $\mathcal{M}$, Dataset tuple $T = (X, S^G)$, mapping function ψ, paraphrasing model θ, evaluation metric $\mathcal{E}$.
initialize $X' = \emptyset$
for each $s \in S^G$ **and** $x \in X$ **pair do**
  let $I_x = \{j | x_j = \psi(s_i), \forall s_i \in s\}$.
  obtain x' by replacing $x_i, \forall i \in I_x$ with $\theta(x_i)$.
  obtain $X' = X' \cup \{x'\}$.
end for
measure $\mathcal{E}(S^G, \mathcal{M}(X))$ **and** $\mathcal{E}(S^G, \mathcal{M}(X'))$.

4 Results

We now present results for assessing robustness through our proposed relevance paraphrasing strategy. We undertake extensive experiments on 4 LLMs of different sizes: GPT-3.5_{Turbo}, Llama-2_{13B}, Mistral_{7B}, and Dolly-$v2_{7B}$, and 4 diverse real-world datasets: CNN/DM (See et al., 2017), XSum (Narayan et al., 2018), Reddit (Kim et al., 2019), and News (Ahmed et al., 2018). We use Llama-2_{13B} as the paraphrasing model for all experiments. Please refer to Appendices A and B for detailed information on the datasets and models, respectively.

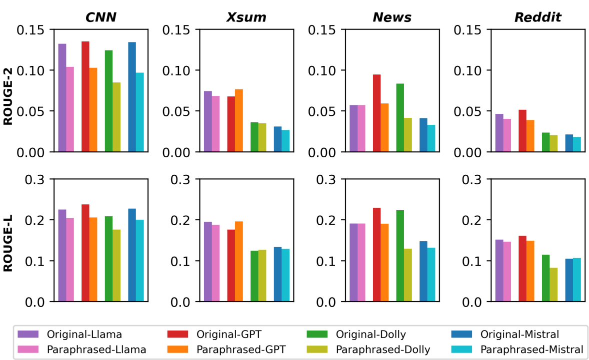

Figure 2: Evaluating summarization performance using ROUGE-2/L on original and paraphrased articles.

Table 1: Performance change (%) observed after relevance paraphrasing across datasets/LLMs.

Datasets	Metrics	Llama-2_{13B}	$\text{GPT-3.5}_{\text{Turbo}}$	Dolly-v2_{7B}	Mistral_{7B}
		Performance Change (%)			
CNN	ROUGE-1	(-)7.354	(-)8.750	(-)13.77	(-)6.814
	ROUGE-2	(-)21.20	(-)23.73	(-)31.66	(-)27.72
	ROUGE-L	(-)9.431	(-)13.54	(-)15.70	(-)11.99
	BertScore	(-)0.311	(-)0.689	(-)5.754	(-)0.522
XSum	ROUGE-1	(-)2.837	(+)16.19	(+)0.680	(-)3.680
	ROUGE-2	(-)8.077	(+)12.99	(-)3.607	(-)13.91
	ROUGE-L	(-)3.764	(+)11.41	(+)1.465	(-)3.649
	BertScore	(-)0.092	(+)0.321	(-)0.524	(+)0.047
News	ROUGE-1	(-)10.90	(-)15.41	(-)39.60	(-)7.457
	ROUGE-2	(-)28.43	(-)36.96	(-)50.30	(-)19.43
	ROUGE-L	(-)13.15	(-)17.00	(-)41.79	(-)10.65
	BertScore	(-)0.080	(-)0.707	(-)7.083	(+)0.528
Reddit	ROUGE-1	(-)3.158	(-)6.600	(-)21.85	(-)2.974
	ROUGE-2	(-)13.10	(-)24.13	(-)13.20	(-)13.89
	ROUGE-L	(-)3.529	(-)7.646	(-)27.64	(-)1.700
	BertScore	(-)0.070	(-)0.750	(-)18.84	(+)2.104

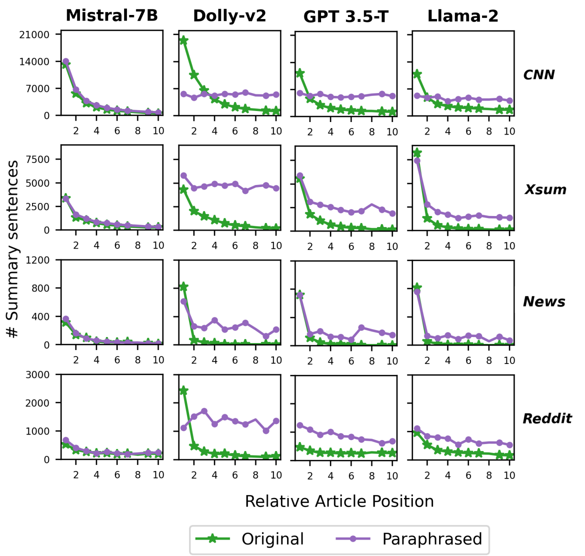

Figure 3: Paraphrasing results in different summaries.

4.1 LLMs Are Not Robust Summarizers

We present the relative performance change[2] (%) for the original LLM summary and the one obtained after relevance paraphrasing in Table 1. We evaluate over 4 holistic summarization metrics: ROUGE-1/2/L and BertScore. We also provide the specific original/paraphrased performance values for the ROUGE-2/L metrics in Figure 2 and defer ones for ROUGE-1 and BertScore showcasing similar trends to Appendix E due to space constraints.

Through these results it can be observed that summarization performance drops significantly after relevance paraphrasing for all LLMs. The largest drops observed are for the CNN/DM and News datasets, of up to 50% on ROUGE-2 for Dolly-v2_{7B}. Moreover, Dolly-v2_{7B} is the most affected by relevance paraphrasing, with significant drops in performance over all datasets. Surprisingly, even $\text{GPT-3.5}_{\text{Turbo}}$ has performance degradation on the minimally perturbed articles, and Mistral_{7B} demonstrates the most robust performance overall. As an exception, $\text{GPT-3.5}_{\text{Turbo}}$ attains large gains in all evaluation metrics after relevance paraphrasing for the XSum dataset. In a few other cases, such as for Mistral (BertScore) and Dolly-v2 (ROUGE), performance has improved post relevance paraphrasing, but only in marginal amounts. These results indicate that *LLMs are not truly robust summarizers, and more improvements need to be made to ensure consistency in outputs.*

4.2 Relevance Paraphrasing Leads to Entirely Different LLM Generated Summaries

We now explore how LLM summarization selection decisions change as a function of relevance paraphrasing. Using our proxy mapping function ψ we can observe the distribution of which input article sentences contributed information to which model summary sentence. In doing so, we can observe these trends for the summaries generated on the original dataset, as well as the minimally perturbed dataset obtained after relevance paraphrasing. These results are shown in Figure 3, and it can be seen that LLMs start utilizing entirely different sentences to generate the summary on the paraphrased input article. While this selection issue is somewhat lesser for Mistral_{7B}, in general, it poses to be a major problem for all other LLMs. These results further strengthen the finding that LLMs are not robust summarizers, as *a minor perturbation in the input space leads to major changes in the output.*

5 Conclusion

In this paper, we propose *relevance paraphrasing* to enable the robustness analysis of LLMs as zero-shot summarizers. Through exhaustive experiments, we find that LLMs are not robust summarizers, and that models begin to use different article sentences to generate summaries for paraphrased articles. Our results indicate that LLMs need further improvements to ensure robustness. By exposing these robustness issues, we believe future work can extend our efforts by proposing *rectification* strategies employed in the instruction finetuning (RLHF) stage[3] that resolve these concerns.

[2] That is, $(new - old)/old * 100$.

[3] As sentences can be paraphrased in multiple ways, doing this in the supervised finetuning stage might be intractable.

Limitations

Our work analyzes the robustness of LLMs as zero-shot summarizers across four diverse datasets. Our results from experiments show that LLMs need to be improved to ensure consistency and robustness in summarization performance (such as via rectification strategies). However, our work has a few limitations that we seek to alleviate in future work. First, summarization robustness needs to assessed in the context of long-form documents (medical records and legal documents, for example) where issues of robustness can lead to adverse outcomes. Second, LLM robustness at summarization needs to be analyzed for low-resource languages and domains where robustness of performance will likely be worsened. Finally, for closed-source models such as GPT-3.5$_{\text{Turbo}}$, a longitudinal analysis of summarization robustness needs to be undertaken, as model performance can change over time.

Ethics Statement

Our work on uncovering summarization robustness issues in LLMs is important to further improve these models, and ensure robustness of performance. A lack of consistency in generating abstractive summaries in a zero-shot setting can lead to adverse outcomes in real-world scenarios, and our results shed light on this issue through experiments on 4 diverse datasets and 4 different LLMs. Through our initial preliminary efforts, we hope to galvanize research efforts to make LLMs more safer and reliable in practice.

References

Hadeer Ahmed, Issa Traore, and Sherif Saad. 2018. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1):e9.

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE Computer Society.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, et al. 2023. Free Dolly: Introducing the world's first truly open instruction-tuned LLM.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Erik Jones, Anca Dragan, Aditi Raghunathan, and Jacob Steinhardt. 2023. Automatically Auditing Large Language Models via Discrete Optimization. *arXiv preprint arXiv:2303.04381*.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive Summarization of Reddit Posts with Multi-level Memory Networks. In *Proceedings of NAACL-HLT*, pages 2519–2531.

Ching-Yun Ko, Pin-Yu Chen, Payel Das, Yung-Sung Chuang, and Luca Daniel. 2023. On Robustness-Accuracy Characterization of Large Language Models using Synthetic Datasets. In *Workshop on Efficient Systems for Foundation Models@ ICML2023*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Fábio Perez and Ian Ribeiro. 2022. Ignore Previous Prompt: Attack Techniques For Language Models. In *NeurIPS ML Safety Workshop*.

Abhinav Rao, Sachin Vashistha, Atharva Naik, Somak Aditya, and Monojit Choudhury. 2023. Tricking LLMs into Disobedience: Understanding, Analyzing, and Preventing Jailbreaks. *arXiv preprint arXiv:2305.14965*.

Joshua Robinson and David Wingate. 2022. Leveraging Large Language Models for Multiple Choice Question Answering. In *The 11th International Conference on Learning Representations*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Chenhui Shen, Liying Cheng, Yang You, and Lidong Bing. 2023. Are Large Language Models Good Evaluators for Abstractive Summarization? *arXiv preprint arXiv:2305.13091*.

Atsushi Shirafuji, Yutaka Watanobe, Takumi Ito, Makoto Morishita, Yuki Nakamura, Yusuke Oda, and Jun Suzuki. 2023. Exploring the Robustness of Large Language Models for Solving Programming Problems. *arXiv preprint arXiv:2306.14583*.

Michal Štefánik. 2022. Methods for Estimating and Improving Robustness of Language Models. In *Proceedings of NAACL-HLT*, pages 44–51.

Jiuding Sun, Chantal Shaib, and Byron C Wallace. 2023. Evaluating the Zero-shot Robustness of Instruction-tuned Language Models. *arXiv preprint arXiv:2306.11270*.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2023. Evaluating the factual consistency of large language models through news summarization. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Priyan Vaithilingam, Tianyi Zhang, and Elena L Glassman. 2022. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *ACM CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–7.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A Multi-Task Benchmark for Robustness Evaluation of Language Models. In *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jiongxiao Wang, Zichen Liu, Keun Hee Park, Muhao Chen, and Chaowei Xiao. 2023. Adversarial Demonstration Attacks on Large Language Models. *arXiv preprint arXiv:2305.14950*.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How Does LLM Safety Training Fail? In *Thirty-seventh Conference on Neural Information Processing Systems*.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023a. A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models. *arXiv preprint arXiv:2303.10420*.

Wentao Ye, Mingfeng Ou, Tianyi Li, Xuetao Ma, Yifan Yanggong, Sai Wu, Jie Fu, Gang Chen, Junbo Zhao, et al. 2023b. Assessing Hidden Risks of LLMs: An Empirical Study on Robustness, Consistency, and Credibility. *arXiv preprint arXiv:2305.10235*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

Chao Zhao, Faeze Brahman, Kaiqiang Song, Wenlin Yao, Dian Yu, and Snigdha Chaturvedi. 2022. NarraSum: A Large-Scale Dataset for Abstractive Narrative Summarization. *arXiv preprint arXiv:2212.01476*.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023a. PromptBench: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. *arXiv preprint arXiv:2306.04528*.

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. 2023b. AutoDAN: Automatic and Interpretable Adversarial Attacks on Large Language Models. *arXiv preprint arXiv:2310.15140*.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

Appendix

A Detailed Dataset Information

CNN/DM (See et al., 2017): The CNN/DM dataset contains 300K news articles written by CNN and Daily Mail employees and journalists. The testing set consists of 11490 articles. The average number of sentences in the articles are 33.37 and on average there are 3.79 sentences per summary.

XSum (Narayan et al., 2018): The XSum dataset contains over 200K short, one-sentence news summaries collected through online articles from the British Broadcasting Corporation. The testing set consists of 11334 articles. The average number of sentences in the articles are 19.105 and on average summaries contain only 1 sentence.

Reddit (Kim et al., 2019): The Reddit dataset consists of 120K Reddit posts where these informal crowd-generated posts constitute the text source, in contrast with existing datasets that use formal documents such as news articles as source. We used an 80-20% train-test split to obtain 4214 articles in the test set. The average number of sentences per article is 22.019 and there are an average of 1.4276 sentences per summary.

News (Ahmed et al., 2018): The News dataset was initially created for fake news classification. We used the testing set comprising of 1000 articles. In the summaries, there are an average number of 1.012 sentences over all articles.

B Detailed Model Information

GPT-3.5$_{\text{Turbo}}$ (Ye et al., 2023a): GPT-3.5-turbo is OpenAI's flagship LLM which has been instruction-tuned and optimized for chat purposes. We utilized the model using the OpenAI API[4] and experiments were conducted on the November version.

Llama-2$_{\text{13B}}$ (Touvron et al., 2023): Meta developed the Llama-2 family of LLMs, a collection of pretrained and fine-tuned generative text models ranging in scale from 7-70 parameters. We use the chat version of the models trained via instruction finetuning. We generated inferences via the PyTorch code provided in the official Github repository: https://github.com/facebookresearch/llama.

Dolly-v2$_{\text{7B}}$ (Conover et al., 2023): Dolly is a 6.9 billion parameter causal language model created by Databricks finetuned on a 15K instruction corpus generated by Databricks employees. We used the *databricks/dolly-v2-7b* checkpoint[5] from HuggingFace as the summarization model.

Mistral$_{\text{7B}}$ (Jiang et al., 2023): This is the first LLM developed by Mistral AI that is a decoder-based model trained with the following architectural choices: grouped query attention, sliding window attention, and byte-fallback tokenization. Due to these choices, despite Mistral$_{\text{7B}}$ being a 7B parameter model, it outperforms Llama-2$_{\text{13B}}$ on a number of evaluation benchmarks.

C Llama-2 Prompts for Paraphrasing

To paraphrase the article sentences that corresponded to the dataset summary sentences we leveraged Llama-2. It is important to note that Llama-2 refused to paraphrase 4.93% of the sentences due to the sentences containing objectionable or problematic language. Therefore we removed all of these articles from both the original and paraphrased datasets before generating the summaries. We now present the prompt used:

You are a helpful assistant that is an expert in paraphrasing sentences. Paraphrase the sentence I will provide. Please respond with just the paraphrased version of the sentence. Here is the sentence: {Sentence}

Note that *{Sentence}* was replaced with the article sentence to obtain the paraphrased sentence. We then replace the original sentence in the article with this version to obtain the minimally perturbed article post relevance paraphrasing.

[4] https://platform.openai.com/docs/models/gpt-3-5

[5] https://huggingface.co/databricks/dolly-v2-7b

D LLM Prompts for Summarization

In this section we provide the prompts used to generate both original and paraphrased summaries for each LLM and each dataset. The number of sentences prompted per dataset is equal to the nearest integer of the average number of sentences in the corresponding gold summaries. The prompts were improved iteratively and tailored to each LLM to ensure the most reliable prompt following. However, sometimes the models did not follow the prompt specifications exactly and would generate more summary sentences than required for that dataset. For e.g. Llama-2 followed the prompt exactly 45.99% while generating the original summaries. Hence, for fair comparison between original and paraphrased summaries we uniformly sampled the number of sentences required from the generated output. We now provide prompts below:

D.1 Prompts for GPT-3.5$_{\text{Turbo}}$

XSum: *For the following article: {Article}. Return a summary comprising of 1 sentence. With each sentence in a numbered list format.*
For example:
1. First sentence

CNN/DM: *For the following article: {Article}. Return a summary comprising of 3 sentences. Write each sentence in a dash bulleted format.*
For example:
1. First sentence
2. Second sentence
3. Third sentence

Reddit: *For the following article: {Article}. Return a summary comprising of 1 sentence. With each sentence in a numbered list format.*
For example:
1. First sentence

News: *For the following article: {Article}. Return a summary comprising of 1 sentence. With each sentence in a numbered list format.*
For example:
1. First sentence

D.2 Prompts for Llama-2$_{\text{13B}}$

XSum: *For the following article: {Article}. Return a summary comprising of 1 sentence. With each sentence in a numbered list format.*

For example:
1. First sentence

CNN/DM: *For the following article: {Article}. Return a summary comprising of 3 sentences. With each sentence in a numbered list format.*
For example:
1. First sentence
2. Second sentence
3. Third sentence

Reddit: *For the following article: {Article}. Return a summary comprising of 1 sentence. With each sentence in a numbered list format.*
For example:
1. First sentence

News: *For the following article: {Article}. Return a summary comprising of 1 sentence. With each sentence in a numbered list format.*
For example:
1. First sentence

D.3 Prompts for Dolly-v2$_{7B}$

XSum: *Generate a 1 sentence summary for the given article. Article: {Article}.*

CNN/DM: *Generate a 3 sentence summary for the given article. Article: {Article}.*

Reddit: *Generate a 1 sentence summary for the given article. Article: {Article}.*

News: *Generate a 1 sentence summary for the given article. Article: {Article}.*

D.4 Prompts for Mistral$_{7B}$

XSum: *For the following article: {Article}. Return a summary comprising of 1 sentence. With each sentence in a numbered list format.*
For example:
1. First sentence

CNN/DM: *For the following article: {Article}. Return a summary comprising of 3 sentences. With each sentence in a numbered list format.*
For example:
1. First sentence
2. Second sentence
3. Third sentence

Reddit: *For the following article: {Article}. Return a summary comprising of 1 sentence. With each sentence in a numbered list format.*
For example:
1. First sentence

News: *For the following article: {Article}. Return a summary comprising of 1 sentence. With each sentence in a numbered list format.*
For example:
1. First sentence

Note that *{Article}* in each prompt should be replaced by the article to be summarized.

E Additional Results on Robustness of LLM Summarization Performance

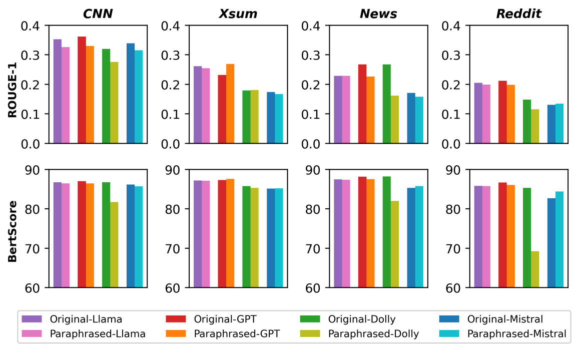

Figure 4: Summarization performance evaluation using ROUGE-1 and BertScore metrics post relevance paraphrasing.

We present results similar to Figure 2 for the BertScore and ROUGE-1 evaluation metrics in Figure 4. It can be seen that for these metrics as well, performance drops consistently across all LLMs post relevance paraphrasing.

F Code and Reproducibility

We open-source our code and provide it as a Github repository: `https://anonymous.4open.science/r/Relevance-Paraphrasing-90BF`. The repository contains instructions for how to reproduce our results and analyze the findings for each model. All the original summaries and articles, as well as the paraphrased articles and summaries for each model and dataset are also provided in this repository for qualitative analysis. We used Python 3.8.10 for all experiments. The experiments were conducted on Ubuntu 20.04 using NVIDIA GeForce RTX A6000 GPUs running with CUDA version 12.0.