Assessing LLMs for Zero-shot Abstractive Summarization Through the Lens of Relevance Paraphrasing

Hadi Askari[†], **Anshuman Chhabra**[§], **Muhao Chen**[†], **Prasant Mohapatra**[§]
[†] Department of Computer Science, University of California, Davis
[§] Department of Computer Science and Engineering, University of South Florida
{haskari,muhchen}@ucdavis.edu, {anshumanc,pmohapatra}@usf.edu

Abstract

Large Language Models (LLMs) have achieved state-of-the-art performance at zero-shot generation of abstractive summaries for given articles. However, little is known about the robustness of such a process of zero-shot 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 model performance for summarization on both the original and perturbed datasets, we can assess the LLM's one aspect of 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 consistent summarizers for the minimally perturbed articles, necessitating further improvements.

1 Introduction

Large Language Models (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 generalizability on discriminative tasks (Wang et al., 2023b, 2022, 2023a; Zhou et al., 2023). One aspect of these tasks is *adversarial robustness* where adversarial prompts meant to induce unsafe behavior are evaluated (Zhu et al., 2023a; Wang et al., 2021). However, we investigate how robust the generative task of abstractive summarization is when the input article is altered via semantic-preserving perturbations. 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.

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*.

To our best knowledge, none of these prior works have explored the robustness of LLM performance at the generative task of *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 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 paraphrasing* 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

Findings of the Association for Computational Linguistics:
NAACL 2025, pages 2187–2201

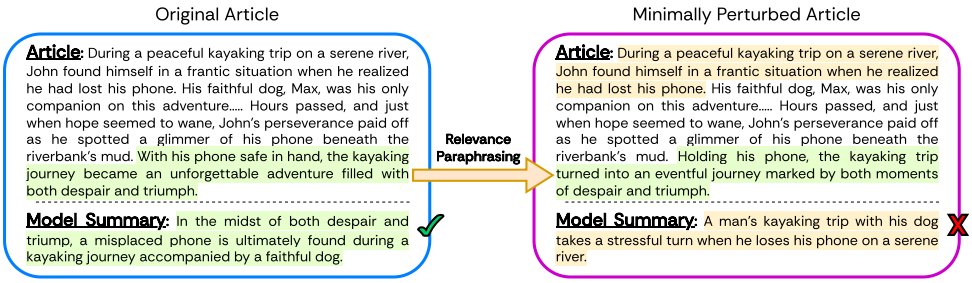

Figure 1: An example showcasing *relevance paraphrasing*. When sentences *relevant* to generating the summary are *paraphrased* to create a minimally perturbed article, we find that summarizaton performance drops as the model uses other sentences instead to craft the summary, leading to a loss of salient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

to the original version but are phrased differently. This gives us a semantically equivalent version of the input set of articles as only a few sentences are paraphrased.[1] 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. [2] Thus, by measuring the change in performance on both the original and perturbed set of input articles, we can assess LLM 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 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[3] as well as multiple datasets. Our results hence indicate that LLMs are not consistent summarizers, and necessitate further improvements to ensure more consistent summarization performance.

2 Measuring Robustness Via Relevance Paraphrasing

2.1 Zero-Shot Summarization

A zero-shot abstractive summarization model $\mathcal{M}$ takes as input X is a set of articles. Each article $x \in X$ has 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 *gold standard* summaries written by human experts (denoted as S^{G}) using evaluation metrics such as ROUGE (Lin, 2004) and BertScore (Zhang et al., 2019).

2.2 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 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.[4] 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.[5] 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.[6] In this paper,

[1] Additional experiments on paraphrasing non-relevant sentences and paraphrasing a larger proportion of the input article are presented in Appendix I and Appendix J.

[2] There are other methods of perturbing inputs in a semantically equivalent manner (e.g. lexical substitution McCarthy and Navigli (2007)). However, we chose relevance paraphrasing as lexical substitution does not introduce much syntactical variance and the introduced perturbation is not sufficient to test summarization robustness.

[3] 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$_{\text{7B-Instruct-v0.1}}$ (Jiang et al., 2023) in all experiments.

[4] We can also return more than one sentence in this framework as described in Appendix M.

[5] Note that other metrics may also be considered. We experimented with ROUGE-1 as an alternative and found no significant differences in the results as shown in Appendix L.

[6] 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.

we use Llama-2_{13B}[7] 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 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.

Algorithm 1 : Relevance Paraphrasing

Input: LLM $\mathcal{M}$, Dataset $T = (X, S^G)$, 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** $\mathcal{E}(S^G, \mathcal{M}(X'))$.

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.[8] 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.

3 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). Please refer to Appendices A and B for detailed information on the datasets and models, respectively.

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

Datasets	Metrics	**Llama-2_{13B}**	**GPT-3.5_{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

3.1 LLMs Are Not Consistent Summarizers

We present the relative performance change[9] 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 provide the specific original/paraphrased performance bar charts, which further elaborate these trends, in Appendix E. We also provide results for LLM based evaluation metrics in Appendix G, using NLI as an evaluation metric in Appendix P, when the temperature parameter set to 0 in Appendix F, traditional summarization models (BART and Pegasus) in Q and results for successive prompting in R.

Through these results it can be observed that summarization performance drops significantly after relevance paraphrasing for all LLMs. The largest drops observed are for CNN/DM and News– up to 50% on ROUGE-2 for Dolly-$v2_{7B}$. Dolly-$v2_{7B}$ is the most affected by relevance paraphrasing, with significant drops in performance over all datasets. Even GPT-3.5_{Turbo} has performance degradation on the minimally perturbed articles, while $Mistral_{7B}$ demonstrates the most robust performance overall. As an exception, GPT-3.5_{Turbo} attains large gains in all evaluation metrics after

[7] We use the instruction-tuned Llama-2 variant throughout.

[8] We perform an automatic evaluation to test the semantic relevance between the original and paraphrased sentences by calculating the BertScore between them– CNN: 0.9387, XSum 0.9410, News: 0.9320 and Reddit: 0.9190, indicating high semantic similarity between the sentences. We also provide a few qualitative examples in Appendix H, along with other paraphrasing models tested in Appendix K.

[9] That is, $(new - old)/old * 100\%$.

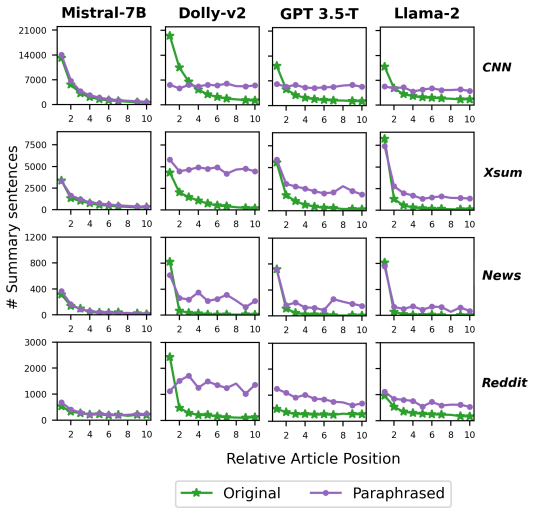

Figure 2: Paraphrasing results in different summaries.

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 show that *LLMs are not consistent summarizers, and more improvements need to be made to ensure consistency in outputs*.

3.2 Relevance Paraphrasing Leads to Mostly Different LLM Generated Summaries

We explore how LLM summarization selection decisions change as a function of relevance paraphrasing. Using our proxy mapping ψ we can observe the distribution of which input article sentences contributed information to which model summary sentence. We then observe these trends pre and post relevance paraphrasing. These results are shown in Figure 2, and it can be seen that LLMs start utilizing 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 consistent summarizers, as *a minor perturbation in the input space leads to significant changes in the output*.

3.3 Human Evaluation of Summaries

To further strengthen our results, we conduct *human evaluation* of different article-summary pairs, along the methodological lines of Zhang et al. (2024); Fabbri et al. (2021). We recruit 22 unpaid student annotators, where each annotator was given up to 30 article-summary pair triplets (gold, original and paraphrased). All 4 LLMs were evaluated on the *XSum* and *Reddit* datasets. Each dataset-model pair was annotated by 3 annotators. The results are provided in Appendix O.

As can be observed, out of the 8 dataset-model pairs, paraphrased summaries were rated as the best only once (1/8 or 12.5% of the time). In contrast, original summaries were preferred 34.5% of the time (3/8) and gold summaries were rated the best 50% (4/8) of the time. It can be seen that relevance paraphrasing leads to summaries that are not as highly preferred (by humans) as the original (non-paraphrased input) and gold summaries.

4 Related Works

LLM robustness has largely been studied in the context of adversarial robustness, where a malicious adversary seeks to execute unsafe model behavior by *automatedly* (Zou et al., 2023; Wang et al., 2023c; 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 to characterize robustness of LLMs at the summarization task. Complementarily, recent work has also shown that LLM based abstractive summarization suffers from position bias, further demonstrating their brittleness at the summarization task (Chhabra et al., 2024).

To our best 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. [10]

5 Conclusion

We propose *relevance paraphrasing* to enable the robustness analysis of LLMs as abstractive summarizers. We find that LLMs are not consistent summarizers, and they begin to use different article sentences to generate summaries for paraphrased

[10] Additional related work for non-LLM neural abstractive summarization and LLM robustness on tasks other than summarization are presented in Appendix N.

articles. Our results indicate that LLMs need further improvements to ensure robustness.

Acknowledgments

Hadi Askari was supported by the NSF Grant ITE 2333736. Muhao Chen was supported by the DARPA FoundSci Grant HR00112490370, the NSF of the United States Grant ITE 2333736, an Amazon Research Award, and an Amazon Trusted AI Prize. Anshuman Chhabra was supported by the USF CSE department faculty startup fund.

Limitations

Our work analyzes the robustness of LLMs as abstractive 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 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 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.

Xiuying Chen, Mingzhe Li, Xin Gao, and Xiangliang Zhang. 2022. Towards improving faithfulness in abstractive summarization. *Advances in Neural Information Processing Systems*, 35:24516–24528.

Yanran Chen and Steffen Eger. 2023. Menli: Robust evaluation metrics from natural language inference. *Transactions of the Association for Computational Linguistics*, 11:804–825.

Yiran Chen, Pengfei Liu, Ming Zhong, Zi-Yi Dou, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Cdevalsumm: An empirical study of cross-dataset evaluation for neural summarization systems. *arXiv preprint arXiv:2010.05139*.

Anshuman Chhabra, Hadi Askari, and Prasant Mohapatra. 2024. Revisiting zero-shot abstractive summarization in the era of large language models from the perspective of position bias. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

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.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Lisa Fan, Dong Yu, and Lu Wang. 2018. Robust neural abstractive summarization systems and evaluation against adversarial information. *arXiv preprint arXiv:1810.06065*.

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*.

Kundan Krishna, Yao Zhao, Jie Ren, Balaji Lakshminarayanan, Jiaming Luo, Mohammad Saleh, and Peter J Liu. 2022. Improving the robustness of summarization models by detecting and removing input noise. *arXiv preprint arXiv:2212.09928*.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*.

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

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Klaus-Michael Lux. 2020. On the factual correctness and robustness of deep abstractive text summarization.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 48–53.

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*.

Arpit Rajauria. 2021. Pegasus paraphrase.

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*.

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*.

Kaiqiang Song. 2021. Towards improving the robustness of neural abstractive summarization. *Doctoral Dissertation, University of Central Florida*.

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*.

Maxim Kuznetsov Vladimir Vorobev. 2023. A paraphrasing model based on chatgpt paraphrases.

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*.

Fei Wang, James Y Huang, Tianyi Yan, Wenxuan Zhou, and Muhao Chen. 2023a. Robust natural language understanding with residual attention debiasing. *arXiv preprint arXiv:2305.17627*.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. 2023b. A causal view of entity bias in (large) language models. *arXiv preprint arXiv:2305.14695*.

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

Yiwei Wang, Muhao Chen, Wenxuan Zhou, Yujun Cai, Yuxuan Liang, Dayiheng Liu, Baosong Yang, Juncheng Liu, and Bryan Hooi. 2022. Should we rely on entity mentions for relation extraction? debiasing relation extraction with counterfactual analysis. *arXiv preprint arXiv:2205.03784*.

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*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

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*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. *arXiv preprint arXiv:2303.11315*.

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[11] 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`. We used the instruction tuned version of Llama-2$_{\text{13B}}$ in all experiments.

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[12] from HuggingFace as the summarization model.

Mistral$_{\text{7B-Instruct-v0.1}}$ (Jiang et al., 2023): This is the first LLM developed by Mistral AI that is a decoder-based model trained with the following

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

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

architectural choices: grouped query attention, sliding window attention, and byte-fallback tokenization. Due to these choices, despite Mistral_{7B} being a 7B parameter model, it outperforms Llama-2_{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.

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 $\text{GPT-3.5}_{\text{Turbo}}$

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_{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

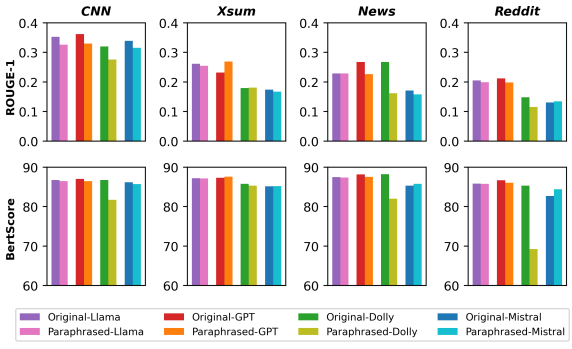

Figure 3: Summarization performance evaluation using ROUGE-1 and BertScore metrics post relevance paraphrasing.

We present results for the BertScore and ROUGE evaluation metrics in Figure 3 and Figure 4. It can be seen that for these metrics as well, performance drops consistently across all LLMs post relevance paraphrasing.

F Temperature 0 Experiments

We re-run the experiment on a 10% sample of the datasets while setting the temperature to zero to further investigate the effects of relevance paraphrasing. Note that we did not perform the main experiments with this setting since general and common usage of Large Language models by practitioners has temperature set to non-zero values for stochasticity in outputs. We wanted to assess the robustness in the more realistic general-purpose use-case scenario. Results are presented in Table 2 and Figure 5. The sentence distribution do not exhibit nearly as much variance as the reference based metrics, this could be a function of the significantly reduced dataset size.

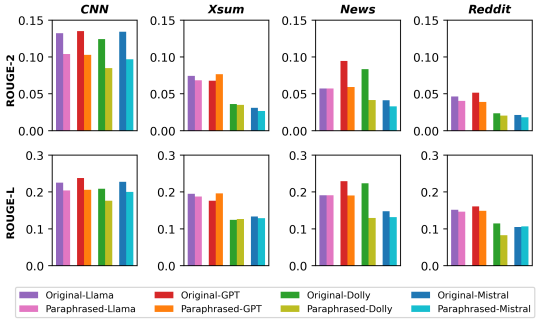

Figure 4: Evaluating summarization performance using ROUGE-2/L on original and paraphrased articles.

Table 2: Performance change (%) observed after relevance paraphrasing across datasets/LLMs with Temperature set to 0.

Datasets	Metrics	Llama-2$_{13B}$	GPT-3.5$_{Turbo}$	Dolly-v2$_{7B}$	Mistral$_{7B}$
		Performance Change (%)			
CNN	ROUGE-1	(-)7.543	(-)5.740	(-)10.45	(-)10.37
	ROUGE-2	(-)20.728	(-)14.74	(-)28.12	(-)28.32
	ROUGE-L	(-)9.842	(-)7.034	(-)13.90	(-)14.33
	BertScore	(-)0.388	(-)0.231	(-)0.353	(-)0.373
XSum	ROUGE-1	(-)5.381	(-)3.448	(-)0.637	(+)1.703
	ROUGE-2	(-)12.405	(-)8.361	(-)4.547	(+)1.414
	ROUGE-L	(-)4.377	(-)3.298	(-)1.420	(+)1.324
	BertScore	(-)0.249	(-)0.125	(-)1.751	(+)0.218
News	ROUGE-1	(+)7.095	(-)2.876	(-)19.37	(-)11.38
	ROUGE-2	(-)3.272	(-)17.79	(-)40.93	(-)34.03
	ROUGE-L	(+)4.312	(-)4.335	(-)22.17	(-)15.59
	BertScore	(+)0.436	(+)0.459	(-)0.331	(-)0.103
Reddit	ROUGE-1	(+)2.174	(-)1.992	(-)23.87	(-)6.461
	ROUGE-2	(-)1.813	(-)9.880	(-)41.31	(-)25.44
	ROUGE-L	(-)0.704	(-)4.075	(-)25.82	(-)6.445
	BertScore	(-)0.020	(-)0.102	(-)22.33	(-)0.053

G G-Eval

To further evaluate the quality of the summaries we leverage LLM based evaluation, G-Eval (Liu et al., 2023). We used GPT-3.5$_{Turbo}$ as the base model and set the temperature to 0. We prompted the LLM-evaluator to re-port back a weighted score from 1 to 5 where 5 is the highest quality. The detailed prompt can be found here Appendix G.1.

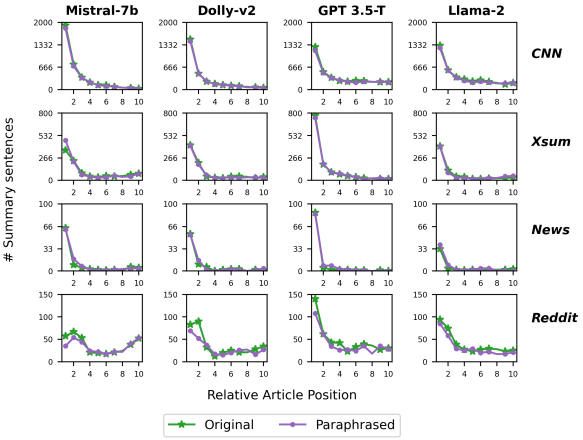

Figure 5: Paraphrasing results in different summaries for Temperature set to 0.

The results are presented in Table 3. The results are inconsistent with the Gold summaries being the highest rated in 9 comparisons, the Original summaries being the highest rated in 3 comparisons and the Paraphrased summaries being the highest rated in 4 comparisons. We find that this lack of consistency in summarization evaluation has been emphasized and observed in past work (Zheng et al., 2024). The identified issues are as follows: LLM-based evaluators suffer from position bias (prefer the first summary over the second), verbosity bias (prefer longer summaries over shorter ones) and self-enhancement bias (preferring outputs generated by themselves) (Zheng et al., 2024). These results may suffer from verbosity bias and self-enhancement bias. The GPT-3.5$_{\text{Turbo}}$ specifically might suffer from self-enhancement bias as the original summaries were generated in September 2023 whereas the paraphrased summaries were generated in December 2023 and OpenAI updates the models regularly.

G.1 G-Eval Prompt:

Prompt: *You will be given one summary written for an article. Your task is to rate the summary based on the following criteria: Output format: PERCENTAGE, PERCENTAGE, PERCENTAGE, PERCENTAGE, PERCENTAGE*

Evaluation Criteria: 1. Read the news article carefully and identify the main topic and key points. 2. Read the summary and compare it to the news article. Check if the summary covers the main topic and key points of the news article, and if it resents them in a clear and logical order. 3. Rate the summary with 5 percentages, where each one represents how likely the summary is going to get a score from 1 to 5. For example, if you think the summary is 80% likely to get a score of 5, 10% likely to get a score of 4, 5% likely to get a score of 3, 3% likely to get a score of 2, and 1% likely to get a score of 1, you should rate the summary as 80, 10, 5, 3, 2.

Here is the article: {Article}

Here is the summary: {Summary}

Table 3: G-Eval (GPT-3.5 Turbo) scores on the three different types of summaries.

Datasets	Summary	Llama-2$_{\text{13B}}$	GPT-3.5$_{\text{Turbo}}$	Dolly-v2$_{\text{7B}}$	Mistral$_{\text{7B}}$
CNN	Gold	4.786	4.786	**4.786**	**4.786**
	Original	**4.811**	4.783	4.765	4.775
	Paraphrased	4.809	**4.814**	4.785	4.751
Xsum	Gold	**4.798**	**4.798**	**4.798**	**4.798**
	Original	4.772	4.721	4.797	4.635
	Paraphrased	4.758	4.728	4.539	4.642
News	Gold	4.753	4.753	4.753	**4.753**
	Original	4.799	4.702	**4.794**	4.648
	Paraphrased	**4.800**	**4.806**	4.784	4.643
Reddit	Gold	4.558	4.558	**4.558**	**4.558**
	Original	**4.745**	4.713	4.348	4.193
	Paraphrased	4.733	**4.782**	4.338	4.208

H Paraphrasing Examples

To further illustrate the paraphrasing done by Llama-2$_{\text{13B}}$ we provide a few examples from the CNN dataset:

- **Original:** They were exposed to Ebola in Sierra Leone in March, but none developed the deadly virus.
- **Paraphrased:** They encountered Ebola in Sierra Leone in March, yet none of them contracted the fatal infection.
- **Original:** The student was identified during an investigation by campus police and the office of student affairs and admitted to placing the noose on the tree early Wednesday, the university said.
- **Paraphrased:** The university reported that the student was discovered by campus police and the office of student affairs to have placed a noose on a tree on Wednesday morning, following an investigation.
- **Original:** Four days after her apparent death, the dog managed to stagger to a nearby farm,

dirt-covered and emaciated, where she was found by a worker who took her to a vet for help.

- **Paraphrased:** The dog, who had been presumed dead for four days, miraculously made her way to a nearby farm, badly injured and severely underweight, where she was discovered by a worker who rushed her to a veterinarian for medical attention.

Additionally, we also provide a few examples from the Reddit dataset that had the highest drop in BertScore between the original and paraphrased article sentences.

- **Original:** I accidentally knocked the cupcakes out of her hand, causing them to spill all over her.
- **Paraphrased:** i knocked the cupcakes out of her hand and they spilled all over her.
- **Original:** i was just now preparing for bed, brushing my teeth, when i reached for my retainer and found it...gone!
- **Paraphrased:** i was getting ready for bed, brushing my teeth, when I realized that my retainer was nowhere to be found.
- **Original:** well as soon as i asked him for a raise, he looked up at me, smiled and pulled out a stack of papers from his desk
- **Paraphrased:** 'As soon as I requested a raise, my boss looked up at me with a smile and pulled out a large pile of papers from his desk.
- **Original:** my 5th period isn't the greatest.
- **Paraphrased:** My fifth period is not particularly outstanding.
- **Original:** this was five years ago (im 18 now), me and my girlfriend were bored and my parents were not at home.'
- **Paraphrased:** 'About five years ago, when I was 18 years old, my girlfriend and I were feeling bored and our parents were not present at home.

I Paraphrasing Non-Relevant Sentences

We generated Paraphrased summaries for the CNN and Reddit datasets on 2000 articles each by randomly paraphrasing the non-relevant sentences. These "non-relevant" sentences were the sentences not selected by our relevance mapping function. We selected an equal amount of non-relevant sentences to paraphrase as we would have selected relevant sentences to paraphrase (for e.g if the summary comprised of 2 sentences, we then paraphrased any two, randomly selected, non-relevant sentences in the input article). We then evaluate and compare with our relevance paraphrasing approach using the Llama-2 LLM.

	CNN		Reddit	
Metrics	**Relevant**	**Random**	**Relevant**	**Random**
ROUGE-1	(-)7.135	(-)1.625	(-)3.880	(-)0.665
ROUGE-2	(-)19.84	(-)3.636	(-)16.17	(-)2.212
ROUGE-L	(-)8.498	(-)1.394	(-)4.178	(-)0.179
BertScore	(-)0.238	(-)0.023	(-)0.078	(+)0.060

We observe that paraphrasing random sentences leads to significantly less effect on the output summaries' quality (as measured by evaluation metrics) compared to our relevance paraphrasing approach. For instance, the drop in ROUGE-1 increases from -7.135 (relevance paraphrasing) to -1.625 (non-relevance paraphrasing) in the CNN dataset and the drop in ROUGE-1 increases from -3.880 (relevance paraphrasing) to -0.665 (non-relevance paraphrasing) in the Reddit dataset. This difference is significant and consistent across the other metrics as well.

J Paraphrasing a larger proportion of the article

We paraphrase the top-3 most relevant sentences (as opposed to only the top-1) on a 2000 subsample of the XSum and News datasets and employ the Llama-2 LLM. This experiment thus seeks to increase the proportion of input article sentences that are being paraphrased and observe the robustness of LLM generated summaries.

	XSum		NEWS	
Metrics	**Top 1**	**Top 3**	**Top 1**	**Top 3**
ROUGE-1	(-)2.483	(-)3.091	(-)10.58	(-)11.21
ROUGE-2	(-)5.926	(-)8.302	(-)28.32	(-)33.22
ROUGE-L	(-)2.644	(-)3.831	(-)12.91	(-)13.64
BertScore	(-)0.050	(-)0.076	(-)0.015	(-)0.022

We can observe some performance degradation

incurred by paraphrasing more sentences. The BertScore drops from -0.050 to -0.076 in XSum and from -0.015 to -0.022 in News. Hence, we can conclude that relevance paraphrasing with a larger proportion of the input space leads to a further slight decrease in summarization performance, while trading off potential semantic similarity with the original article. We would like to emphasize that the original relevance paraphrasing approach applied minimal perturbation while providing similar evidence of summarization robustness.

K Other Paraphrasing Models

We investigate other non-LLM paraphrasing models "Pegasus Paraphrase" (Rajauria, 2021) and "chatgpt paraphraser on T5 base" (Vladimir Vorobev, 2023), since they were popular and well performing on Huggingface. We eventually settled on using Llama-2_{13B} as the paraphraser as we saw that the quality of the outputs was significantly better.

To quantify this, we paraphrased 10 sentences each with these 3 models and calculated the BertScore and ROUGE-1 score with the original sentence.

Table 4: Sampled ROUGE-1 and Bertscore scores on different paraphrasing models.

Model	Bertscore	ROUGE-1
Pegasus Paraphrase	94.29	80.17
chatgpt paraphraser on T5 base	93.92	72.20
Llama-2_{13B}	92.16	60.64

Qualitatively, we analyze a few examples to demonstrate:

- **Original:** Zarif left the country in 1977, received his undergraduate degree from San Francisco State University in 1981, his master's in international relations from the University of Denver in 1984 and his doctorate from the University of Denver in 1988.
- **"Pegasus Paraphrase":** Zarif received his undergraduate degree from San Francisco State University in 1981 and his master's in international relations from the University of Denver in 1984 while he was still in the country.
- **"chatgpt paraphraser on T5 base":** In 1977, Zarif departed from the country, but in 1981, he received his undergraduate degree from San Francisco State University, a master's degree in international relations from the University of Denver, and a doctorate from the University of Denver in 1988.
- **Llama-2_{13B}**: Zarif departed the country in 1977, later earning his undergraduate degree from San Francisco State University in 1981, his master's in international relations from the University of Denver in 1984, and his doctorate from the same institution in 1988.

Here, "Pegasus Paraphrase" omits the detail that Zarif left the country in 1977 and "chatgpt paraphraser on T5 base" omits the detail of when he received his masters. Llama2-13B, on the other hand, keeps all the pertinent details while paraphrasing as well.

Looking at another example:

- **Original:** Looking spry at 91, Barker handled the first price-guessing game of the show, the classic "Lucky Seven," before turning hosting duties over to Carey, who finished up.
- **"Pegasus Paraphrase":** Barker handled the first price-guessing game of the show, the classic "Lucky Seven," before handing hosting duties to Carey, who finished up.
- **"chatgpt paraphraser on T5 base":** The first game of the show, the classic "Lucky Seven," was handled by Barker, who then handed over the hosting duties to Carey, who finished up the show looking sappy at 91.
- **Llama-2_{13B}**: Barker, who appeared youthful at 91, kicked off the first price-guessing game of the show, "Lucky Seven," before passing the baton to Carey to conclude the segment.

Here, "Pegasus Paraphrase" again misses the detail that Barker was looking spry at 91 years of age. "chatgpt paraphraser on T5 base" misrepresents the information and paraphrases that Carey was the one who is 91 when actually it was Barker. Finally, Llama2-13B paraphrases the sentence elegantly.

L Additional Results for Different ψ

For experiments in the main paper, we opt for TF-IDF vector similarities as the choice of the mapping function ψ due to computational efficiency (over computing individual ROUGE scores between summary and article sentences

for e.g.). However, it is important to examine whether this choice significantly impacts results, trends, and our findings. In initial experiments with different ψ we concluded that this choice does not affect results. In Figure 6 we provide results that support this by using ROUGE-1 as the metric for ψ on the *Reddit* and *News* datasets for Llama-2 generated summaries. We compare the gold summary and original summary positional distributions for both datasets when ψ is computed using TF-IDF vectors and ROUGE-1. It is clear that the trends and results are the same for both ψ.

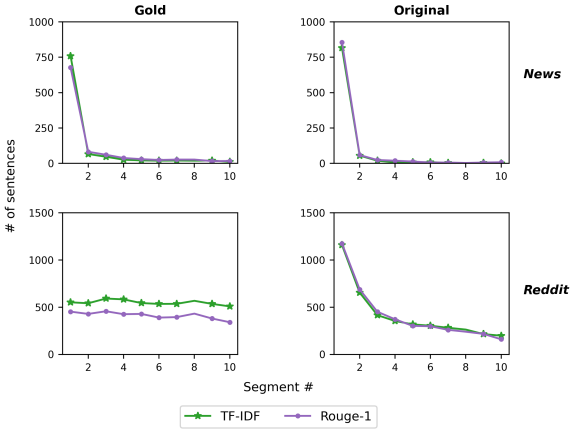

Figure 6: Results on *News* and *Reddit* for Llama-2 when ψ is either TF-IDF similarity or ROUGE-1.

M Mapping Summary Sentences to Top-N relevant Article Sentences

Currently, ψ maps back from one summary sentence to one article sentence that contributes the most to that summary sentence. To do this, as ψ measures similarity between sentences, we currently only pick the article sentence with the maximum similarity to the summary sentence. However, since ψ is basically measuring similarity, we can return the top-2 or top-3 matches and undertake the same sentence distribution analysis as in the main results. No specific change is necessary, since our sentence distribution estimation is done in aggregate, via binning. It can be seen that the distributions do change slightly, but overall the trends remain the same. It is beneficial to assess the impact of utilizing multiple article sentences, especially for datasets like *XSum* where the summary is usually just one sentence and discusses facts from multiple article sentences.

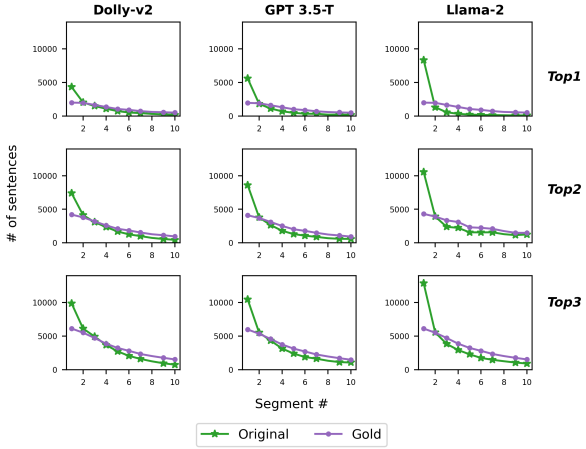

Figure 7: Mapping summary sentences to multiple article sentences on *XSum*.

N Additional Related Work

N.1 LLM Robustness on tasks other than Summarization

Other work on LLM robustness has proposed evaluation workflows to assess model performance at general instruction following (Sun et al., 2023), program synthesis (Shirafuji et al., 2023), sentence classification (Ko et al., 2023), reasoning problems (Ye et al., 2023b), enhancing generalizability via debiasing outputs (Wang et al., 2023a, 2022, 2023b) and mitigating prediction shortcuts (Zhou et al., 2023).

N.2 Summarization Robustness for non-LLMs

Prior work has also studied the robustness of non-LLM neural abstractive summarization models. However, these works have traditionally focused on summary factuality/faithfulness as a proxy for measuring robustness (Song, 2021; Fan et al., 2018; Chen et al., 2022). For instance, in (Song, 2021) the authors propose encoding structural information in summarization models and in (Fan et al., 2018; Chen et al., 2022), the authors propose new architectures/models to improve factual robustness of generated summaries. Note that the research question we address in our work is distinct from this line of work as our goal is to identify if minor perturbations of the input space can lead the LLMs to generate different summaries (both the original and paraphrased summaries could still be faithful/factual to the input article). Other past work (Chen et al., 2020; Lux, 2020; Krishna et al., 2022) has investigated robustness of non-LLM summarizers in distribution shift scenarios by assessing how models

trained on a particular dataset perform on other datasets (Chen et al., 2020), analyzing the temporal issue where articles are changing over time (Lux, 2020), and adding synthetic data pipeline / text extraction noise to the input to assess performance (Krishna et al., 2022). As can be seen, our paper considers a somewhat related but overall separate problem where we only perturb very few sentences to retain semantic similarity ("relevance paraphrasing"), and then observe if that changes the sentence selection process the LLM uses for summarizing and subsequent summary quality. Past work has not utilized relevance paraphrasing in this manner for robustness analysis possibly due to the general brittleness issues of non-LLM summarization models (Kryściński et al., 2019).

O Human Evaluation

The following instructions were provided to the annotators[13] (similar protocol as to (Zhang et al., 2024): *Annotate the summary quality of the following Article-summary pairs using the definitions. Your scores will be presented in a research article. Definitions:*

- ***Faithfulness (Binary Score 0 or 1):*** *Whether the summary is faithful to the article or not (on topic).*
- ***Coherence (Integer Score between 1 to 5):*** *Whether the summary is well-structured and organized. Should not just be a heap of related information*
- ***Relevance (Integer Score between 1 to 5):*** *Whether the summary selects important content from the source. Summary should only include the important info.*

The results obtained are as follows. For each dataset-model pair the best summary (in bold) is the one with overall high performance on the 3 sub-metrics (to break ties we use the overall highest scores):

Gold			
Dataset-Model	**Faithfulness**	**Coherence**	**Relevance**
Xsum GPT	0.520(0.242)	3.390(1.257)	2.280(0.440)
Xsum Llama	0.550(0.201)	2.870(0.640)	2.380(0.779)
Xsum Dolly	0.630(0.163)	3.580(0.984)	2.820(0.505)
Xsum Mistral	0.630(0.123)	3.470(0.456)	3.340(0.560)
Reddit GPT	0.720(0.077)	2.750(0.622)	2.820(0.645)
Reddit Llama	0.670(0.133)	2.950(0.572)	2.950(0.230)
Reddit Dolly	0.830(0.095)	3.100(0.272)	3.620(0.308)
Reddit Mistral	0.790(0.160)	3.090(1.057)	3.270(0.500)

Original			
Dataset-Model	**Faithfulness**	**Coherence**	**Relevance**
Xsum GPT	0.950(0.077)	4.000(0.514)	4.380(0.375)
Xsum Llama	0.990(0.030)	3.520(0.600)	3.680(0.380)
Xsum Dolly	0.670(0.116)	3.370(0.552)	2.670(0.289)
Xsum Mistral	0.770(0.177)	3.270(0.668)	2.800(0.608)
Reddit GPT	1.000(0.000)	4.220(0.816)	4.400(0.492)
Reddit Llama	0.920(0.101)	3.850(0.358)	3.900(0.232)
Reddit Dolly	0.520(0.248)	2.340(0.534)	2.070(0.315)
Reddit Mistral	0.400(0.077)	3.100(0.439)	2.220(0.477)

Paraphrased			
Dataset-Model	**Faithfulness**	**Coherence**	**Relevance**
Xsum GPT	0.920(0.073)	3.760(0.545)	3.720(0.146)
Xsum Llama	0.990(0.030)	3.550(0.303)	3.590(0.305)
Xsum Dolly	0.820(0.186)	3.370(0.622)	2.720(0.305)
Xsum Mistral	0.820(0.085)	3.250(0.648)	2.870(0.590)
Reddit GPT	0.890(0.030)	4.130(0.497)	3.920(0.581)
Reddit Llama	0.950(0.071)	3.880(0.363)	4.120(0.093)
Reddit Dolly	0.500(0.179)	2.230(0.737)	2.040(0.506)
Reddit Mistral	0.370(0.173)	2.930(0.509)	1.950(0.469)

P NLI based Evaluation

We use MENLI (Chen and Eger, 2023) to perform NLI evaluation of our summaries as it is a recent work introducing metrics for robust NLI. We sub-sample the CNN, XSum and Reddit datasets to 2k samples each and evaluate on the llama2 outputs. For MENLI, our evaluation parameter settings were: Direction=rh, formula='e-c', nli_weight=0.3, combine_with="BERTScore-F", model="D" in line with the best performing configuration for summarization in their paper.

	Llama-2$_{13B}$
Dataset	**% change**
CNN	(-)0.566
XSum	(-)1.193
News	(-)3.226
Reddit	(-)23.53

We can observe a consistent degradation in the quality of outputs that are in-line with the findings of our paper. The performance change is negative throughout and the drop in performance is highest on Reddit (-23.53%) and News (-3.266%). This further emphasizes the effect of our relevance paraphrasing to analyze the robustness of LLM summarization.

[13]The annotators were grad students of UC Davis residing in Davis, California.

Q Results for Traditional Summarization Models BART and Pegasus

We provide robustness results after *relevance paraphrasing* for the *fine-tuned* versions of BART and Pegasus.

Dataset	Metric	BART	Pegasus
CNN	Rouge1	(-)13.04	(-)14.1
	Rouge2	(-)33.71	(-)34.61
	RougeL	(-)18.38	(-)19.73
	BertScore	(-)0.676	(-)0.828
Xsum	Rouge1	(-)3.14	(-)2.978
	Rouge2	(-)5.601	(-)4.836
	RougeL	(-)3.53	(-)3.139
	BertScore	(-)0.22	(-)0.218
News	Rouge1	(-)10.71	(-)12.3
	Rouge2	(-)21.9	(-)22.71
	RougeL	(-)11.6	(-)12.64
	BertScore	(-)0.587	(-)0.763
Reddit	Rouge1	(-)5.4	(-)8.81
	Rouge2	(-)15.41	(-)22.43
	RougeL	(-)6.263	(-)10.15
	BertScore	(-)0.137	(-)0.38

Table 5: Robustness degradation after relevance paraphrasing for BART and Pegasus.

We can clearly see a robustness degradation across the board. For example, BART shows a higher drop in quality than Llama2-13b-chat across all metrics and datasets, except for the ROUGE metrics in the News dataset. However, this is expected, as humans tend to prefer LLM-generated summaries over those produced by supervised models. Hence, poorer robustness is a consequence of poorer overall performance. Therefore, we recommend using LLMs for abstractive summarization in general. However, even LLMs need to be made more robust (i.e., they are not as resilient to relevance paraphrasing as human summarizers) despite their better performance compared to BART and Pegasus.

R Successive Prompting of Original Articles

We perform successive prompting and compare the results with the results in Table 1. We can clearly see that there is significantly less change in the outputs and that it is also in not as uniform a degradation. A similar comparison can be made with Table 2.

S Code and Reproducibility

We open-source our code and provide it as a Github repository: `https://anonymous.4open.science/r/Relevance-Paraphrasing-D1EB/`.

Table 6: Performance change (%) observed after prompting the LLM with the same article summary pair twice

Datasets	Metrics	**Llama-2$_{13B}$**	**GPT-3.5$_{Turbo}$**	**Dolly-v2$_{7B}$**	**Mistral$_{7B}$**
		Performance Change (%)			
CNN	ROUGE-1	(-)0.039	(+)0.328	(-)0.119	(+)0.014
	ROUGE-2	(-)0.088	(+)0.909	(+)0.016	(+)0.134
	ROUGE-L	(-)0.013	(+)0.294	(-)0.040	(-)0.029
	BertScore	(+)4.66E-08	(+)0.009	(+)2.60E-08	(+)7.75E-08
XSum	ROUGE-1	(-)0.014	(+)0.347	(+)0.069	(-)0.033
	ROUGE-2	(-)0.033	(-)1.622	(+)0.049	(-)0.329
	ROUGE-L	(+)0.096	(+)0.128	(-)0.052	(-)0.046
	BertScore	(+)9.91E-08	(+)0.005	(+)1.33E-07	(+)7.43E-08
News	ROUGE-1	(+)0.146	(+)1.202	(+)0.552	(-)0.141
	ROUGE-2	(+)0.530	(-)1.919	(-)0.137	(-)0.343
	ROUGE-L	(+)0.218	(-)0.268	(+)0.431	(+)0.185
	BertScore	(+)0.000	(+)0.059	(+)5.86E-08	(+)1.17E-07
Reddit	ROUGE-1	(-)0.047	(-)2.392	(-)0.160	(+)0.032
	ROUGE-2	(-)0.375	(-)3.429	(-)0.654	(-)0.757
	ROUGE-L	(+)0.035	(-)2.060	(-)0.393	(-)0.004
	BertScore	(-)2.60E-07	(+)0.043	(+)2.60E-07	(-)9.00E-08

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.