Beyond Static Synthetic Noise: Assessing the Robustness of Large Language Models to Natural Context Variation in the Real World

Yulong Wu[1], Viktor Schlegel[1,2] and Riza Batista-Navarro[1]
[1] Department of Computer Science, University of Manchester, United Kingdom
[2] Imperial College London, Imperial Global Singapore
{yulong.wu, riza.batista}@manchester.ac.uk
v.schlegel@imperial.ac.uk

Abstract

Robustness evaluation in Question Answering (QA) has predominantly relied on synthetic perturbations that poorly capture natural text evolution in real-world settings, a limitation that becomes more pronounced with the widespread deployment of Large Language Models (LLMs) in dynamic, user-facing environments. In this work, we address this gap by proposing a framework for automatically evaluating QA models under naturally occurring textual perturbations, replacing context passages with revised counterparts from Wikipedia edit histories. Through extensive evaluation on SQuAD across diverse encoder architectures, we construct two challenging sets where human performance remains stable, yet state-of-the-art LLMs exhibit significant degradation, with performance drops of up to 28.28%. These robustness gaps further generalise to more complex QA scenarios, such as DROP and HotpotQA. To mitigate these errors, we show that robustness to natural perturbations can be improved via adversarial training for encoder-only models and in-context demonstrations of perturbed instances for LLMs, though a more generalisable and effective defense strategy remains an open challenge[1].

1 Introduction

Large Language Models (LLMs) increasingly serve as core components of autonomous agents and Retrieval-Augmented Generation (RAG) systems, requiring reasoning over external knowledge sources rather than static inputs (Lewis et al., 2020; Luo et al., 2025). In real-world deployments, these sources, such as Wikipedia or live knowledge bases, are continuously updated through human edits, resulting in naturally evolving textual contexts (Yang et al., 2017). Consequently, system reliability depends not only on reasoning ability but also on robustness to text drift–the natural variations introduced as content is revised, expanded, or reorganized over time.

Current robustness evaluation research, however, lags behind this reality (Wang et al., 2022; Zhang et al., 2025). Most existing approaches rely on static synthetic perturbations based on predefined manipulation strategies to stress-test models (Wu et al., 2023a; Gupta et al., 2024; Bhuiya et al., 2024). While these methods offer insights into model sensitivity under controlled conditions, they act as artificial proxies that do not reflect the unintentional and semantically coherent changes characteristic of real-world text evolution (Figure 1). As a result, current evaluation protocols risk overlooking practical vulnerabilities, leaving agents underprepared for real-world deployments where grounding must remain robust under naturally evolving contexts (Wu et al., 2025).

To counteract this issue, we introduce a framework for evaluating language model robustness under naturally occurring textual changes in real-world settings. We focus on Question Answering (QA), which serves as a high-fidelity proxy for how agents and RAG systems ground their reasoning in retrieved text. Following the intuition of (Belinkov and Bisk, 2018), we leverage Wikipedia revision histories as a source of natural perturbations, since the differences between revisions authentically capture the textual modifications made by human editors in the real world. By comparing adjacent revisions, we construct perturbed versions of context passages in existing QA benchmarks when available, while keeping questions and ground-truth answers unchanged.

With the established framework, we conduct extensive experiments on six datasets, evaluating forty-two models, including recently proposed LLMs. Experimental results on Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016, 2018) indicate that natural pertur-

[1]Our code and the constructed challenge dataset are available at https://github.com/Yulong-W/natural_perturbation.

Findings of the Association for Computational Linguistics: ACL 2026, pages 36050–36070
July 2-7, 2026

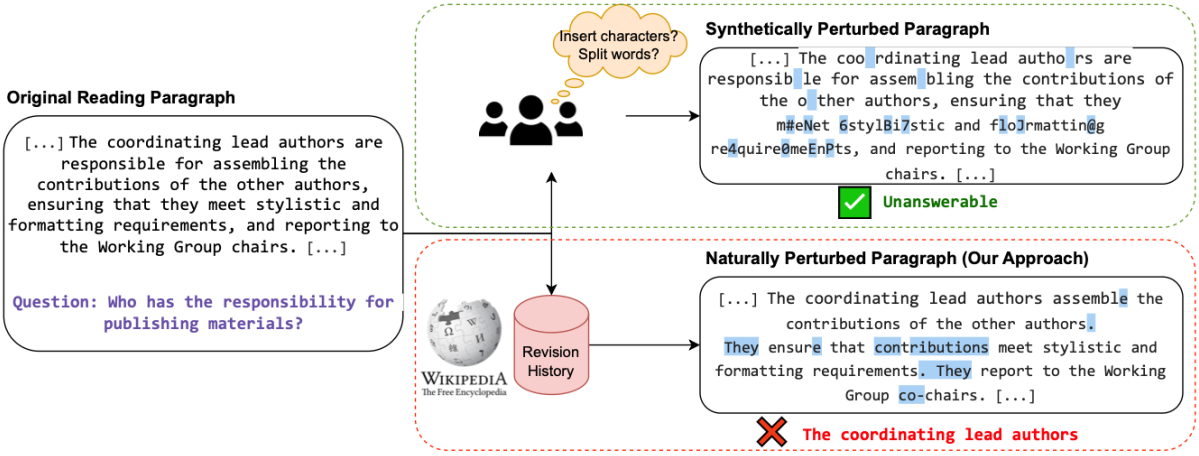

Figure 1: Given a reading paragraph, we extract and use Wikipedia revision history to construct its naturally perturbed version for a more realistic robustness evaluation (Bottom), rather than relying on a set of synthetic methods (Top). While `Mistral-7B-Instruct-v0.2` generates the correct answer for both the original and synthetically perturbed passages, it fails under natural perturbation.

bations encompass rich linguistic variations and can lead to failures in the encoder-only models, while humans are almost undeterred by their presence. Crucially, these errors also transfer to larger and more powerful models, such as `Flan-T5` and SOTA LLMs, with performance drops ranging from 4.4% to 28.28%. These findings also generalise to other and more challenging QA benchmarks (e.g., `Mistral-7B-Instruct-v0.2`'s 5.25% decrease on BoolQ (Clark et al., 2019) and `Llama-3.2-1B-Instruct`'s 9.98% decline on DROP (Dua et al., 2019)), emphasising the harmful effects of natural perturbations. Adversarial re-training/in-context demonstration with either naturally or synthetically perturbed QA instances can enhance the robustness against natural perturbations, with the latter sometimes providing greater benefits. However, there is still ample room for improvement, calling for better defense strategies.

The contributions of this paper are as follows:

- A framework—based on Wikipedia revision history—for studying model robustness under real-world natural perturbations. This is relevant, even in the LLM era, as our framework can be applied to any other tasks with input from Wikipedia and also to any types of models.
- Perturbed datasets for six diverse QA tasks. Two SQUAD challenge sets derived from error analysis of encoder-only models, on which SOTA LLMs struggle, even without being involved in the creation in any capacity.
- Empirical demonstration of the validity of natural perturbations across both encoder-only models and LLMs, their characterisation by different linguistic phenomena and their harmful effects on diverse model architectures across benchmarks generated with the proposed framework.
- Showcasing adversarial re-training with natural or, especially, synthetic perturbations, as well as adversarial in-context demonstrations as a way to enhance the robustness of encoder-only models and LLMs, respectively, against natural perturbations.

2 Natural Perturbation Pipeline

We design a pipeline to automatically construct label-preserving stress QA test sets with noises that occur in real-world settings by leveraging Wikipedia revision histories (Figure 2). Our approach comprises two modules: *candidate passage pairs curation* and *perturbed test set construction*.

Candidate passage pairs curation. For each English Wikipedia article within the development set[2] of QA datasets, we systematically extract its entire revision histories and preprocess them, including the removal of markups and the segmentation of content. Subsequently, we obtain the content differences between each current revision and the previous adjacent one, identifying three distinct editing

[2]Since not all test sets are public, we apply natural perturbations to the development sets. For simplicity, we use the term "test set" throughout.

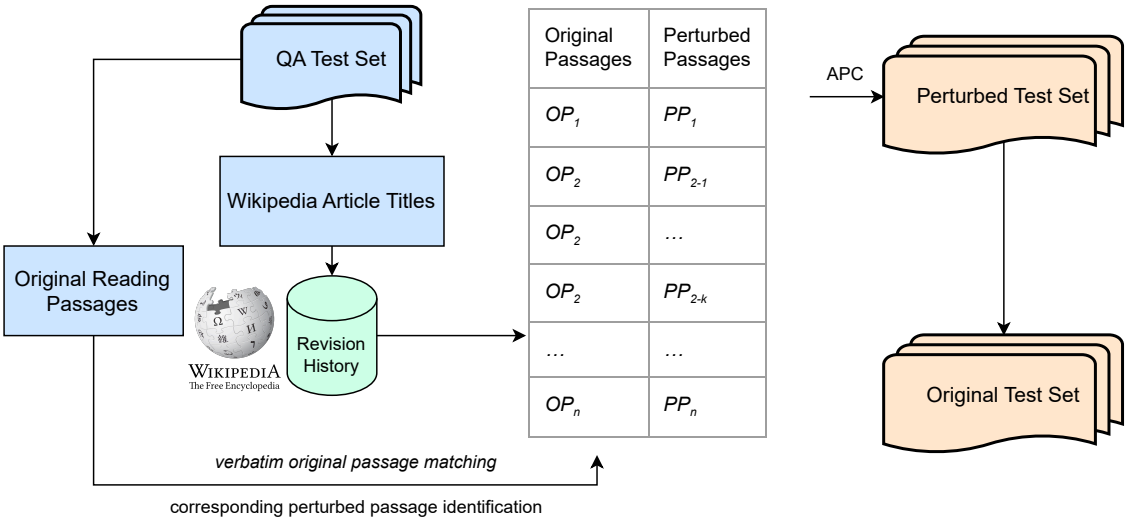

Figure 2: Process of generating naturally perturbed QA test sets. APC: Answers Preserving Checking.

patterns: addition, deletion, and modification. In the case of an edit falling within the modification pattern, we retain the paragraph from the prior version as the *original* and the corresponding one from the current version as the *perturbed*, provided both paragraphs exceed 500 characters[3].

Perturbed test set construction. To generate the naturally perturbed test set, we begin by acquiring all reading passages from the development set of each QA dataset and identifying their entries in the collection of previously extracted candidate original passages, along with the corresponding perturbed counterparts. Subsequently, for the matched original passages with a single occurrence, we keep them and the corresponding perturbed passages; whereas for those with multiple occurrences, we randomly select one instance for each and extract its perturbed version. After obtaining the perturbed reading passages, we retain only those with at least one question where all annotated ground truth answers (or all plausible answers for the unanswerable question) can still be located within the perturbed context, resulting in the *Perturbed* test set. For the sake of comparison, we also construct an *Original* version of the test set keeping only the original passages and questions corresponding to those that were included in the *Perturbed* version.

[3]This threshold setting adheres to the methodology employed in the collection of SQuAD 1.1 (Rajpurkar et al., 2016).

3 Experiments Setup

Our experimental design addresses the following question: *How well do modern language models perform on QA under real-world natural perturbations?* To this end, we first establish a baseline evaluation using encoder-only models on SQuAD. This choice is motivated by the dataset's simplicity, the stable and super-human performance of encoder-only models, and its resistance to benchmark leakage–factors that enable a focused and controlled examination of perturbation effects, error sources, and instance validity. Then, we evaluate the cross-model transferability of these errors to more advanced architectures, specifically `FLAN-T5` and SOTA LLMs. Finally, we generalise the findings from the baseline evaluation to more complex, non-extractive QA scenarios to assess the broader impact of natural perturbations. Below we outline and motivate the choices of datasets, models, and evaluation metrics used in the study.

Datasets: We use six English QA datasets: SQuAD 1.1 (Rajpurkar et al., 2016), SQuAD 2.0 (Rajpurkar et al., 2018), BoolQ (Clark et al., 2019), DROP (Dua et al., 2019), HotpotQA (distractor) (Yang et al., 2018) and TyDi QA (gold passage task in English) (Clark et al., 2020). These are chosen as their reading passages are sourced from Wikipedia, thereby enabling the utilisation of Wikipedia editing histories to generate the naturally perturbed test set.

Models: Our evaluation study involves QA models across three different types: encoder-only, encoder-decoder, and decoder-only. Access to and experimentation with all models are possible via the use of the HuggingFace's *Transformers* library (Wolf et al., 2020), the vLLM library (Kwon et al., 2023), two 80GB Nvidia A100 GPUs and the OpenAI ChatGPT API.

Encoder-only: We select `BERT` (Devlin et al., 2019) and its various variants for evaluation, including `DistilBERT` (Sanh et al., 2019), `SpanBERT` (Joshi et al., 2020), `RoBERTa` (Liu et al., 2019), `ALBERT` (Lan et al., 2020) and `DeBERTa` (He et al., 2021). We fine-tune these encoder-only pre-trained language models on the training set of the two SQuAD datasets and evaluate them on the constructed original and perturbed test sets. Model details and the hyperparameters used in model fine-tuning are shown in Appendix A.

Encoder–Decoder: Instruction finetuning has been demonstrated to be effective in enhancing zero-shot performance of pretrained language models, resulting in the development of Finetuned Language Net (FLAN) (Wei et al., 2022). In this work, we use the instruction-finetuned version of T5 model class, specifically the `Flan-T5` (Chung et al., 2022), available in sizes ranging from *small* (80M), *base* (250M), *large* (780M) to *xl* (3B). During evaluation, we utilise the instruction templates from QA task collection in open-sourced FLAN repository and report the model performance as the average of those obtained across the employed templates. Refer to Appendix B for various instruction templates used for the evaluation on the test sets with the format as the two SQuAD datasets.

Decoder-only: There is an exponential increase of pre-trained generative LLMs and their fine-tuned chat versions, inspired by the remarkable success of ChatGPT (Bang et al., 2023). Therefore, our experiments incorporate a broad range of recently proposed language model families, including `GPT 3.5 Turbo`, `GPT-4o` (OpenAI et al., 2024), `Gemma` (Mesnard et al., 2024), `Gemma 2` (Riviere et al., 2024), `Llama 2` (Touvron et al., 2023), `Llama 3` and `Llama 3.1` (Dubey et al., 2024), `Llama 3.2`, `Mistral` (Jiang et al., 2023), `OLMo` (Groeneveld et al., 2024), `Qwen2.5` (Qwen et al., 2025), `Falcon` (Almazrouei et al., 2023), `Falcon3` (Team, 2024), and `DeepSeek LLM` (DeepSeek-AI et al., 2024). The zero-shot prompts designed for soliciting their responses are presented in Appendix C.

Evaluation Metrics: In line with existing literature, we choose the (instance-averaged) Token-F1 score to assess the performance of both encoder-only and encoder-decoder models (Rajpurkar et al., 2016), as on SQuAD-style test sets, they are optimised to output the shortest continuous span from the context as the answer (or predict the question as unanswerable) during inference. However, the outputs of the decoder-only models do not consistently adhere to the instruction due to their conversational style, rendering F1 unsuitable for evaluation. Consequently, we employ a more lenient metric, namely Inclusion Match (IM), which measures whether the response of the model contains any of the ground truth answers (Bhuiya et al., 2024)[4]. Furthermore, if the model's output includes phrases such as "I cannot answer this/the question" or "unanswerable"[5], we deem that the model believes the question is not answerable. Model robustness is quantified by measuring the relative variation in performance (as reflected in the F1 or IM) under natural perturbations.

4 Results and Analysis

Here, we present and discuss the findings of our study.

Natural perturbations derived from Wikipedia editing histories lead to performance degradation in encoder-only QA architectures on* SQuAD*, with these models also demonstrating considerable robustness. This is evidenced in Table 1, where all examined models show a decline in performance; however, the drops are relatively modest, with the largest F1 reduction being only 3.06%.

A diverse range of edit types is observed within natural perturbations, covering phenomena that previous synthetic approaches do not, yet no single edit type is uniquely predictive of model failure. Within the original and the naturally perturbed test set pair generated based on SQuAD 2.0 development set, we first identify 384 instances where at least one encoder-only model succeeds on the original but fails on the perturbed (i.e., being adversarial), and then randomly select the same number

[4]While the IM may appear overly permissive, a manual inspection of a 468-sample set indicates that the precision for IM-positive and IM-negative labels is 99.8% and 95.1%, respectively, demonstrating the reliability of this metric.

[5]We collate a collection of such phrases by manually examining the decoder-only models' outputs (Check Appendix D for the full set).

Model	SQuAD 1.1	SQuAD 2.0	
		Overall	(Ans./Unans.)
distilbert-base	−0.6	−0.71	(−2.76/1.71)
bert-base-cased	−0.21	−0.63	(−1.84/0.6)
bert-base-uncased	−0.87	−0.49	(−1.88/0.94)
bert-large-cased	−0.63	−0.53	(−1.61/0.55)
bert-large-uncased	−0.35	−1.38	(−2.51/−0.24)
spanbert-base-cased	−0.26	−1.24	(−2.66/0.15)
spanbert-large-cased	−0.51	−1.20	(−1.9/−0.56)
roberta-base	−0.61	−0.60	(−2.09/0.81)
roberta-large	−0.29	−1.52	(−2.6/−0.54)
albert-base-v1	−1.0	−1.07	(−2.02/−0.22)
albert-base-v2	−0.34	−1.08	(−2.03/−0.22)
albert-large-v1	−0.42	−0.41	(−1.42/0.52)
albert-large-v2	−0.8	−0.69	(−1.66/0.22)
albert-xxlarge-v1	−0.75	−1.23	(−3.06/0.49)
albert-xxlarge-v2	−0.46	−1.28	(−3.02/0.36)
deberta-large	−0.52	−1.05	(−2.2/0.0)

Table 1: Relative F1 change (%) for encoder-only QA systems subjecting to natural perturbations. For SQuAD 2.0, the overall values are broken down to answerable and unanswerable questions, respectively.

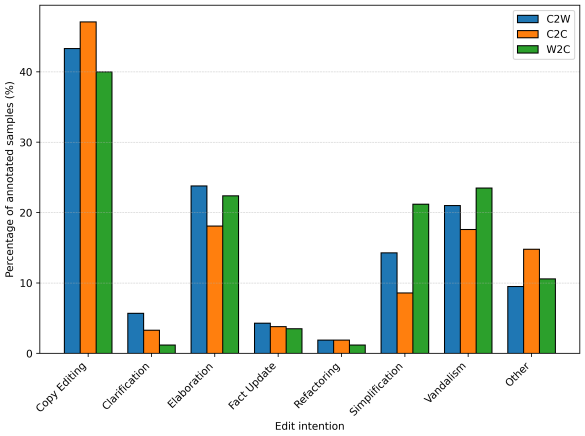

Figure 3: The percentage (%) of samples annotated with each edit intention in the C2W, C2C and W2C categories. The percentages do not add up to 100% because a single revision may fall into multiple intentions.

of instances on which all encoder-only models succeed on both the original and perturbed versions (Naik et al., 2018). We refer to these two types of instances as C2W (correct to wrong) and C2C (correct to correct) instances, respectively. Among the identified C2W and C2C instances, we further remove duplicates, resulting in 210 and 244 unique original and perturbed paragraph pairs, respectively. Furthermore, as natural perturbation can occasionally help the model to get the answer correct, we also filter 85 unique W2C (wrong to correct) instances on which at least two encoder-only models fail on the original but succeed on the perturbed. Finally, utilising an 8-category taxonomy of the semantic edit intentions in Wikipedia revisions derived from Yang et al. (2017), the chosen 210 samples of C2W and C2C, as well as the 85 W2C were annotated, with 20% of the annotated C2W and C2C examples presented to a second annotator for additional validation. See Appendix E for the instruction provided to the annotators, along with detailed explanations of each edit intention. We calculate the (micro-averaged) F1 score to evaluate the inter-annotator agreement, which is 0.82. This suggests that the annotators' annotations align closely. Figure 3 reports the annotation results.

Distribution of perturbation types shown in Figure 3 generally aligns with the edit intentions distribution annotated in (Yang et al., 2017), with *Copy Editing* and *Elaboration* appearing more frequently than others, such as *Clarification*, *Fact Update*, and *Refactoring*. This reflects the inherent characteristics of Wikipedia revisions. From Figure 3, we observe that there is no significant difference in the distribution of annotated edit intentions between C2W and C2C examples, suggesting that though these types of natural perturbations confuse the encoder-only QA models, there seems no correlation with human-perceivable features. A roughly similar distribution is also observed in the W2C examples, which indicates that these natural perturbation types can also facilitate correct answers by the models, i.e., being beneficial. These demonstrate that on SQuAD 2.0, there might be no correlation between the quality of the naturally perturbed passage and its potential for being adversarial[6]. Certain text edits aimed at improving the passage quality, such as *Copy Editing* and *Elaboration*, do render the perturbation adversarial, whereas edits intended to damage the article may not consistently result in adversarial instances; in fact, *vandalism* can even assist models in providing correct answers. Instead, we infer that whether an edit to the passage can render the QA instance adversarial or not depends on the location of the edits in relation to the question. Among the 384 C2W and C2C examples, we measure the proportion of answerable questions with the answer sentence(s) in the original passage remaining unmodified in the naturally perturbed version, which is 34.5% and 71.5%, respectively. This confirms our hypothesis that if the edits affect the answer sentence(s), there is a higher

[6]We also find little or no significant correlation between the perturbation magnitude (measured as byte-level changes between the original and perturbed passages) and model failure, with point biserial correlation coefficient close to 0.

likelihood of the perturbed example becoming adversarial; otherwise, it might not. *Copy Editing* appears to alter the answer sentences in the reading passage more frequently, making it the most impactful category that confuses models (contributing to more than 40% of error cases), while other types have a lesser effect. Appendix F presents one perturbed example for each of the C2W, C2C, and W2C categories, respectively, along with the annotated natural perturbation type(s).

A significant majority of natural adversarial samples are valid, offering superiority over synthetic methods in maintaining human answerability. Three human annotators are recruited to evaluate the 210 C2W examples for validity using the methodology detailed in Appendix G, reaching a high inter-annotator agreement (0.77 Cohen's κ) and confirming that 86% of these instances remain human-answerable. A manual audit of the remaining 14% (29) cases that failed human validation reveals that 89.7% may stem from the inherent difficulty of the QA rather than the perturbations themselves; specifically, all these are unanswerable questions that humans might fail even when presented with the clean, original passage. Within these invalid cases, the distribution of edit intentions mirrors the global distribution shown in Figure 3–with *Copy Editing* contributes to the majority (48.3%), followed by *Elaboration* and *Simplification* (20.7%), *Other* (13.8%), *Vandalism* (10.3%), *Clarification* (6.9%), *Fact Update* (3.4%). Instances of incorrect fact updates or potential misinformation involvement, which occur primarily within the *Fact Update*, account for only 3.4% of invalid cases, further evidencing the validity of natural perturbations. Moreover, this validity rate compares favorably with existing synthetic adversarial approaches, which frequently suffer from unnatural linguistic perturbations or label inconsistencies. See Appendix H for a comparative evaluation of multiple synthetic perturbation methods.

Errors induced by natural perturbations in encoder-only models generalize to more advanced* `FLAN-T5` *and SOTA LLMs. Reflecting recent advancements in SOTA Natural Language Processing (NLP), we stress-test `FLAN-T5` and a diverse suite of recent LLMs using two challenge benchmarks–NAT_V1_CHALLENGE (184 contexts, 234 questions) and NAT_V2_CHALLENGE (214 contexts, 442 questions (226 unanswerable))–which are constructed by extensively harvesting failure cases from encoder-only models (Table 2). Details of the exhaustive search algorithm used to construct the challenge sets are provided in Appendix I.

From Table 2, we observe that the errors caused by natural perturbation in encoder-only MRC models transfer to both `Flan-T5` and LLMs. On the NAT_V1_CHALLENGE, `flan-t5-small` demonstrates the greatest susceptibility to natural perturbations, experiencing a 14.27% decrease in F1; while among LLMs, `Gemma-7B-IT` emerges as the least robust, with a 16.66% IM drop, followed by `Gemma-2B-IT` (−15.83%) and `Llama-3.1-8B-Instruct` (−15.61%). Transitioning to the NAT_V2_CHALLENGE, the base version of `flan-t5` exhibits the largest performance decline at 13.83% and `Falcon-7B-Instruct` stands out as the LLM with the lowest robustness (−28.28%). Other LLMs such as `Qwen2.5-7B-Instruct` and `deepseek-llm-7b-chat` also show severe robustness loss, with drops of 12.21% and 11.29%, respectively. Further, we observe that the robustness of models under natural perturbations does not necessarily size-dependent. While larger models tend to exhibit greater robustness in some cases (e.g., `Qwen2.5-14B-Instruct` vs. `Qwen2.5-3B-Instruct`), exceptions within the `Falcon` and `Llama` model series suggest that factors beyond model size–such as training corpora, training and fine-tuning methodology, and architectural differences may also significantly affect susceptibility to natural perturbations. In Appendix J, we showcase two adversarial examples targeting LLMs sourced from our generated challenge sets.

The threat of natural perturbations extends to more complex QA scenarios. The two SQUAD investigated previously are relatively simple, as they lack challenging features (Schlegel et al., 2020), leading to super-human performance of QA models (Lan et al., 2020). To generalise our findings to more challenging QA scenarios, we apply the natural perturbation methodology (Section 2) to the development set of four more datasets and assess the performance changes of multiple LLMs, as shown in Table 3. For DROP (Dua et al., 2019), we first use the `GPT-4o mini` to infer the likely Wikipedia article title from which each passage is retrieved [7] and extract the revision histories for

[7] This is because the raw Wikipedia title information cannot be found in the original development set of DROP. We use the prompt: "Given a reading paragraph, return the Wikipedia page title from which it is likely retrieved."

Model	Performance *original* vs. *perturbed*			
	NAT_V1_CHALLENGE		NAT_V2_CHALLENGE	
flan-t5-small	58.76/64.76	48.58/55.52$_{-14.27}$	42.57/44.57	39.71/41.81$_{-6.19}$
flan-t5-base	79.49/85.01	66.1/73.42$_{-13.63}$	70.66/72.85	61.16/62.78$_{-13.83}$
flan-t5-large	88.1/92.53	76.57/82.31$_{-11.05}$	79.11/81.01	70.14/72.13$_{-10.96}$
flan-t5-xl	86.25/91.57	75.0/81.45$_{-11.05}$	83.71/85.84	73.19/74.86$_{-12.79}$
GPT-3.5-turbo-0125	91.03	83.33$_{-8.46}$	51.58	47.06$_{-8.76}$
gpt-4o-2024-11-20	93.16	85.9$_{-7.79}$	80.09	75.11$_{-6.22}$
Gemma-2B-IT	51.28	43.16$_{-15.83}$	55.66	50.23$_{-9.76}$
Gemma-7B-IT	82.05	68.38$_{-16.66}$	59.95	57.01$_{-4.9}$
Gemma 2-2b-IT	85.47	78.21$_{-8.49}$	48.87	43.44$_{-11.11}$
Gemma 2-9b-IT	89.32	81.62$_{-8.62}$	64.93	59.95$_{-7.67}$
Llama 2-chat-7B	82.91	73.93$_{-10.83}$	41.63	38.69$_{-7.06}$
Llama 2-chat-13B	80.77	73.93$_{-8.47}$	46.83	41.18$_{-12.06}$
Llama-3-8B-Instruct	88.89	77.35$_{-12.98}$	51.81	46.61$_{-10.04}$
Llama-3.1-8B-Instruct	87.61	73.93$_{-15.61}$	61.31	55.43$_{-9.59}$
Llama-3.2-1B-Instruct	54.27	47.86$_{-11.81}$	35.29	32.13$_{-8.95}$
Llama-3.2-3B-Instruct	81.2	71.37$_{-12.11}$	48.42	43.44$_{-10.29}$
Mistral-7B-Instruct-v0.2	84.19	73.08$_{-13.2}$	54.98	51.36$_{-6.58}$
OLMo-7B-0724-Instruct	90.17	82.91$_{-8.05}$	51.36	49.1$_{-4.4}$
Qwen2.5-3B-Instruct	78.63	68.38$_{-13.04}$	61.31	54.07$_{-11.81}$
Qwen2.5-7B-Instruct	88.03	81.2$_{-7.76}$	76.02	66.74$_{-12.21}$
Qwen2.5-14B-Instruct	92.31	81.62$_{-11.58}$	80.54	74.21$_{-7.86}$
Falcon-7B-Instruct	53.42	50.00$_{-6.4}$	32.81	23.53$_{-28.28}$
Falcon-40B-Instruct	69.66	62.82$_{-9.82}$	38.69	36.88$_{-4.68}$
Falcon3-7B-Instruct	88.03	79.49$_{-9.7}$	59.28	55.43$_{-6.49}$
Falcon3-10B-Instruct	90.6	82.91$_{-8.49}$	64.48	59.73$_{-7.37}$
deepseek-llm-7b-chat	70.51	64.1$_{-9.09}$	42.08	37.33$_{-11.29}$

Table 2: Performance (%) of Flan-T5 and SOTA LLMs on NAT_V1_CHALLENGE and NAT_V2_CHALLENGE. Values in smaller font are changes (%) relative to the original performance of the model.

those articles. For HOTPOTQA (Yang et al., 2018), we only perturb the paragraphs containing the supporting facts, while the distracting passages remain unchanged. We also manually verify the validity percentage of all adversarial examples in DROP and TYDI QA, as well as 50 randomly selected adversarial examples from BOOLQ and HOTPOTQA, as reported in Table 3.

LLM	IM Relative Change (%)			
	BOOLQ	DROP	HOTPOTQA	TYDI QA
adversarial validity (%) (Cohen's κ)	72 (0.54)	85.7 (0.46)	88 (0.6)	87.5 (0.52)
Gemma 2-2b-IT	−3.91	−2.22	–	−1.61
Gemma 2-9b-IT	−3.92	−1.69	−2.21	−1.51
Llama-3.1-8B-Instruct	−3.05	−7.13	−0.91	3.17
Llama-3.2-1B-Instruct	−3.81	−9.98	−1.73	−9.1
Llama-3.2-3B-Instruct	−3.74	–	−2.05	–
Mistral-7B-Instruct-v0.2	−5.25	−1.85	−1.16	–
OLMo-7B-0724-Instruct	−4.49	−7.9	−2.36	2.94
Qwen2.5-7B-Instruct	−4.24	−3.85	−1.18	−2.78
Qwen2.5-14B-Instruct	−3.22	–	−1.84	1.38
Falcon3-7B-Instruct	−5.1	2.04	−0.06	−8.82
Falcon3-10B-Instruct	−3.58	1.8	−1.81	−4.22

Table 3: IM changes (%) of SOTA LLMs on naturally perturbed test set of other more challenging QA datasets.

Overall, when natural perturbations are applied to other more challenging benchmarks, SOTA LLMs also exhibit a lack of robustness, with the largest 9.98 performance decrease observed for `Llama-3.2-1B-Instruct` on DROP. This further demonstrates the broad and severe impact of natural perturbations on diverse QA tasks, especially in light of on three out of four benchmarks, our human annotators are still able to correctly answer over 85% of the adversarial examples. BOOLQ exhibits a lower adversarial validity rate (72%). However, in most cases, this is not due to the perturbation degrading the passage, but rather the poor quality of the original instance (Tedeschi et al., 2023), i.e., even with the corresponding original passage, humans are unable to assign the correct label. For instance, from the annotators' perspective, the question itself may be ambiguous, such as *"do you need a visa to visit oman"*, or entirely unanswerable due to missing information in the original passage. Our human annotations also observe diverse error patterns in LLMs caused by natural perturbations (e.g., *Copy Editing*, *Elaboration* and *Vandalism*), as what are presented in Figure 3.

5 Dealing With Natural Perturbations

In this section, we provide an initial exploration of methods to defend against natural perturbations. To enhance encoder-only model robustness, we first conduct adversarial training by identifying six encoder-only model architectures that already exhibit the highest robustness to natural perturbations in their respective categories (except `albert-xxlarge-v2` on NAT_V2_CHALLENGE), and presenting them with both original training data and the generated naturally perturbed training examples. We extract the entire Wikipedia revision histories for the 392 articles in the original SQUAD training set, and then obtain 5,262 (with 22,033 questions) and 5,311 (with 32,993 questions) perturbed contexts to augment the original SQUAD 1.1 and SQUAD 2.0 training set, respectively, using the methodology described in Section 2. Table 4 compares the performance of these models on NAT_V1_CHALLENGE and NAT_V2_CHALLENGE, before and after retraining.

Apart from re-training with the same type of noise, we also ask whether exposing models to synthetic perturbations can help them confront natural ones. Therefore, we incorporate thirteen synthetic perturbation techniques spanning character and word levels (see Table 9 in Appendix H). Afterwards, we first retrain `deberta-large` with perturbed training samples generated by each synthetic perturbation method, respectively, and assess the performance changes compared to the vanilla version on both NAT_V1_CHALLENGE and NAT_V2_CHALLENGE (Figure 8 in Appendix K). As we observe that synthetic adversarial training can assist `deberta-large` in handling natural per-

Model	Performance (EM/F1) *original* vs. *perturbed*			
	NAT_V1_CHALLENGE		NAT_V2_CHALLENGE	
distilbert-base	64.53/70.45	41.03/47.6$_{-32.43}$	56.56/59.08	41.18/43.3$_{-26.71}$
	57.26/63.44	43.59/51.87$_{-18.24}$	53.17/55.4	43.89/45.51$_{-17.85}$
bert-large-cased	79.06/83.66	63.68/70.23$_{-16.05}$	66.29/68.35	53.17/55.04$_{-19.47}$
	74.79/80.14	59.83/67.5$_{-15.77}$	67.87/69.31	58.37/59.53$_{-14.11}$
spanbert-large-cased	84.19/88.2	67.95/74.77$_{-15.23}$	78.73/80.68	62.44/64.99$_{-19.45}$
	82.48/86.6	69.66/76.05$_{-12.18}$	78.28/80.0	65.61/67.12$_{-16.1}$
roberta-large	86.75/90.21	73.93/79.47$_{-11.91}$	82.13/84.27	66.29/68.52$_{-18.69}$
	83.33/87.15	70.94/76.53$_{-12.19}$	81.22/82.67	70.59/71.84$_{-13.1}$
albert-xxlarge-v2	84.62/89.64	73.93/78.77$_{-12.13}$	84.62/86.07	68.1/69.61$_{-19.12}$
	86.32/90.93	75.64/81.07$_{-10.84}$	82.58/84.08	70.59/72.78$_{-13.44}$
deberta-large	88.46/92.5	73.5/78.48$_{-15.16}$	85.07/86.65	71.49/73.0$_{-15.75}$
	88.03/91.84	76.92/81.53$_{-11.23}$	83.03/85.1	72.62/74.48$_{-12.48}$

Table 4: Comparison of the performance of several encoder-only QA systems on NAT_V1_CHALLENGE and NAT_V2_CHALLENGE, before and after retraining. The results shown in the shaded areas represent the performance of the model retrained on the augmented training set with naturally perturbed instances.

turbations, we further retrain five other models in the same manner and quantify the performance difference on NAT_V1_CHALLENGE compared to the vanilla version, as shown in Figure 4.

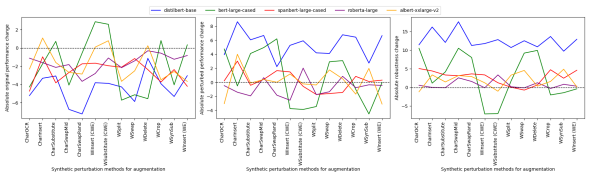

Figure 4: Absolute changes in original and perturbed performance (F1), as well as the robustness of five encoder-only models under natural perturbations (on NAT_V1_CHALLENGE), following retraining with each synthetic perturbation.

In general, for encoder-only QA models, retraining with natural perturbations enhances the performance on naturally perturbed test sets and improves the robustness to such perturbations as well, though this can lead to varying reductions in performance on the clean test set. Encouragingly, adversarial training with synthetically perturbed examples benefits the model's capability to handle natural perturbations as well, a phenomenon differs from what is reported in machine translation task (Belinkov and Bisk, 2018). In some cases, the improvement even exceeds what achieved by retraining the model on natural perturbations alone. We also observe that the effectiveness of adversarial training varies with model size and architecture. Generally, adversarial training brings the most significant benefits for the weakest `distilbert-base`, with the benefits diminishing in larger and more complex model architectures.

Similarly, for the LLMs, we adopt a few-shot prompting approach by including both the original QA instance and its naturally or synthetically perturbed counterpart as demonstrations, and assess how model performance and robustness change compared to the zero-shot setting (see Table 5 and Table 6). A total of two original–perturbed instance pairs are used, with the original samples taken from the SQUAD training set. Although not widely observed, in certain cases, in-context demonstrations can improve an LLM's resilience to natural perturbations, regardless of whether natural or synthetic perturbed examples are demonstrated. This phenomenon is particularly evident in models such as `Llama-3.2-3B-Instruct`, `OLMo-7B-0724-Instruct` and `Falcon3-10B-Instruct`. However, it can also have detrimental effects, further decreasing LLM robustness and resulting in a performance decline on both the clean and naturally perturbed test sets.

LLM	NAT_V1_CHALLENGE			NAT_V2_CHALLENGE		
	Orig./Pert.	*IM Drop*	*zero-shot*	*Orig./Pert.*	*IM Drop*	*zero-shot*
Gemma 2-2b-IT	80.77/72.22 ↓	−10.59	−8.49	45.93/42.53 ↓	−7.4	−11.11
Gemma 2-9b-IT	↑ 91.88/79.91 ↓	−13.03	−8.62	62.9/57.24 ↓	−9.0	−7.67
Llama-3.1-8B-Instruct	↓ 85.04/74.79 ↑	−12.05	−15.61	45.25/41.86 ↓	−7.49	−9.59
Llama-3.2-3B-Instruct	70.09/62.82 ↓	−10.37	−12.11	43.67/43.44 ↓	−0.53	−10.29
Mistral-7B-Instruct-v0.2	↓ 83.33/77.78 ↑	−6.66	−13.2	50.45/46.61 ↓	−7.61	−6.58
OLMo-7B-0724-Instruct	76.5/76.07 ↓	−0.56	−8.05	53.62/51.58 ↑	−3.8	−4.4
Qwen2.5-3B-Instruct	61.11/47.44 ↓	−22.37	−13.04	67.42/61.54 ↑	−8.72	−11.81
Qwen2.5-7B-Instruct	86.32/73.08 ↓	−15.34	−7.76	76.24/69.46 ↑	−8.89	−12.21
Qwen2.5-14B-Instruct	85.47/72.65 ↓	−15.0	−11.58	80.09/73.76 ↓	−7.9	−7.86
Falcon3-7B-Instruct	86.32/75.21 ↓	−12.87	−9.7	↑ 60.86/54.98 ↓	−9.66	−6.49
Falcon3-10B-Instruct	85.9/79.06 ↓	−7.96	−8.49	61.54/59.5 ↓	−3.31	−7.37
deepseek-llm-7b-chat	↑ 70.94/55.98 ↓	−21.09	−9.09	59.73/51.58 ↑	−13.64	−11.29

Table 5: Performance and IM drop of LLMs in the few-shot setting with both original and *naturally* perturbed QA instances demonstrated. *zero-shot* represents the IM drop in the zero-shot setting, adopted from Table 2. Results that evidence robustness improvement in the few-shot setting are underlined.

LLM	NAT_V1_CHALLENGE			NAT_V2_CHALLENGE		
	Orig./Pert.	*IM Drop*	*zero-shot*	*Orig./Pert.*	*IM Drop*	*zero-shot*
Gemma 2-2b-IT	82.48/72.22 ↓	−12.44	−8.49	47.29/42.53 ↓	−10.07	−11.11
Gemma 2-9b-IT	↑ 91.45/81.62	−10.75	−8.62	61.54/56.33 ↓	−8.47	−7.67
Llama-3.1-8B-Instruct	83.33/72.65 ↓	−12.82	−15.61	49.32/44.57 ↓	−9.63	−9.59
Llama-3.2-3B-Instruct	66.24/62.82 ↓	−5.16	−12.11	42.53/41.86 ↓	−1.58	−10.29
Mistral-7B-Instruct-v0.2	↓ 81.62/76.92 ↑	−5.76	−13.2	50.68/46.61 ↓	−8.03	−6.58
OLMo-7B-0724-Instruct	73.84/75.53 ↓	2.29	−8.05	54.98/53.39 ↑	−2.89	−4.4
Qwen2.5-3B-Instruct	59.4/51.71 ↓	−12.95	−13.04	66.52/61.09 ↑	−8.16	−11.81
Qwen2.5-7B-Instruct	85.04/73.5 ↓	−13.57	−7.76	76.47/69.0 ↑	−9.77	−12.21
Qwen2.5-14B-Instruct	84.19/73.5 ↓	−12.7	−11.58	↑ 81.9/74.21	−9.39	−7.86
Falcon3-7B-Instruct	85.47/74.36 ↓	−13.0	−9.7	61.09/56.56 ↑	−7.42	−6.49
Falcon3-10B-Instruct	85.9/79.91 ↓	−6.97	−8.49	61.31/57.92 ↓	−5.53	−7.37
deepseek-llm-7b-chat	↑ 72.22/56.41 ↓	−21.89	−9.09	61.09/53.62 ↑	−12.23	−11.29

Table 6: Performance and IM drop of LLMs in the few-shot setting with both original and *synthetically* perturbed QA instances demonstrated. *zero-shot* represents the IM drop in the zero-shot setting, adopted from Table 2. Results that evidence robustness improvement in the few-shot setting are underlined.

6 Related Work

Robustness Evaluation in QA. A typical approach to evaluate the robustness of QA models

is via test-time perturbation. This line of research develops different perturbation methods as attacks, such as adversarial distracting sentence addition (Jia and Liang, 2017; Tran et al., 2023), explicit connectives removal (Wu et al., 2021), entity renaming (Yan et al., 2022) and paraphrasing (Gan and Ng, 2019; Lai et al., 2021; Wu et al., 2023a). Our work also fits within the category of test-time perturbation, but differs from previous works in that we introduce perturbations that naturally occur in real-world scenarios, therefore contributing to a more practical robustness test.

Natural Perturbation for Robustness Assessment. Compared with deliberately crafting the perturbed instances, the study of natural perturbation is quite under-explored. In the computer vision domain, researchers find that real-world clean images without intentional modifications can confuse deep learning models as well, terming them as natural adversarial examples (Hendrycks et al., 2021; Pedraza et al., 2022). Similarly, in the field of NLP, naturally occurring perturbations extracted from human-written texts can also degrade model performance in tasks such as machine translation (Belinkov and Bisk, 2018) and toxic comments detection (Le et al., 2022). Motivated by these, we attempt to harvest natural perturbations from available Wikipedia revision histories and utilise them to modify the original QA instances. To the best of our knowledge, we are the first to investigate QA model robustness under real-world natural perturbations.

7 Conclusion

In this paper, we first study the robustness of QA models to *natural* perturbations, which occur under real-world conditions without intentional human intervention. Using the proposed evaluation framework, we show that certain naturally perturbed examples can indeed be adversarial, i.e., lead to model failure, even when the modifications aim to improve the overall passage quality. Natural perturbations also appear to differ significantly from synthetic ones, exhibiting a wide range of rich linguistic phenomena and may be more effective in generating valid adversarial instances. Adversarial training via augmentation with either naturally or synthetically perturbed samples is generally beneficial for enhancing the model's robustness to natural perturbations; yet, it can decrease performance on clean test set. Future work includes the exploration of alternative natural perturbation approaches and the design of more effective defensive strategies.

Limitations

We acknowledge several limitations in our work: (1) Our perturbation framework constructs natural perturbations from Wikipedia edit history and therefore only works with Wikipedia-based benchmarks. Since the phenomenon of natural perturbations is by no means limited to Wikipedia and can occur in any kind of text that evolves over time, future work should explore alternative methods to generate natural perturbations for non-Wikipedia QA datasets. (2) As training data augmentation and in-context demonstration have a relatively limited impact, further research is needed to develop better techniques for improving the robustness of both encoder-only models and LLMs to natural perturbations, and to investigate the relationship between robustness to natural and synthetic perturbations. (3) Potential benchmark contamination may affect our findings on LLM evaluation. Investigating its extent and impact on LLM performance and robustness evaluation will be a focus of our future research efforts.

Ethical Considerations

All datasets, extracted natural perturbations, and models used in this work are publicly available, used consistently with their intended purpose and under the permitted license. A very small proportion of natural perturbations may contain offensive content, as they come from reverted Wikipedia revisions intended to damage the articles. We include these to raise awareness within the community about their potential impact on QA models and to call for methods to improve the safety of QA models–especially those LLMs operating under such adversarial conditions. While our ultimate goal is to enhance model robustness, the findings from this work may carry the risk of being misused by malicious attackers to refine adversarial attack strategies and craft attacks against similar systems. Before starting the annotation task, we provide all annotators with clear instructions and inform the intended use of their annotations, obtaining their explicit consent. No private or sensitive information was collected, other than their annotations.

Acknowledgements

We thank the financial support from the Kilburn Scholarship from the Department of Computer Science, The University of Manchester, in making this research possible. We also thank the responsible reviewers from the ARR 2026 January cycle and other previous cycles for their constructive feedback, which helped improve this paper. Experiments were conducted using the Computational Shared Facility at The University of Manchester, for which we are grateful.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *Preprint*, arXiv:2311.16867.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Neeladri Bhuiya, Viktor Schlegel, and Stefan Winkler. 2024. Seemingly plausible distractors in multi-hop reasoning: Are large language models attentive readers? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2514–2528, Miami, Florida, USA. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, and 69 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *Preprint*, arXiv:2401.02954.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 516 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya

Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Ashim Gupta, Rishanth Rajendhran, Nathan Stringham, Vivek Srikumar, and Ana Marasovic. 2024. Whispers of doubt amidst echoes of triumph in NLP robustness. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5533–5590, Mexico City, Mexico. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. 2021. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Diptesh Kanojia, Marina Fomicheva, Tharindu Ranasinghe, Frédéric Blain, Constantin Orăsan, and Lucia Specia. 2021. Pushing the right buttons: Adversarial evaluation of quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 625–638, Online. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 989–1002, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Thai Le, Jooyoung Lee, Kevin Yen, Yifan Hu, and Dongwon Lee. 2022. Perturbations in the wild: Leveraging human-written text perturbations for realistic adversarial attack and defense. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2953–2965, Dublin, Ireland. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, and 7 others. 2025. Large language model agent: A survey on methodology, applications and challenges. *Preprint*, arXiv:2503.21460.

Gemma Team Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, L. Sifre, Morgane Riviere, Mihir Kale, J Christopher Love, Pouya Dehghani Tafti, L'eonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Am'elie H'eliou, Andrea Tacchetti, and 88 others. 2024. Gemma: Open models based on gemini research and technology. *ArXiv*, abs/2403.08295.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference.

In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China. Association for Computational Linguistics.

Anibal Pedraza, Oscar Deniz, and Gloria Bueno. 2022. Really natural adversarial examples. *International Journal of Machine Learning and Cybernetics*, 13(4):1065–1077.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram'e, Johan Ferret, Peter Liu, Pouya Dehghani Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 176 others. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*.

Viktor Schlegel, Marco Valentino, Andre Freitas, Goran Nenadic, and Riza Batista-Navarro. 2020. A framework for evaluation of machine reading comprehension gold standards. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5359–5369, Marseille, France. European Language Resources Association.

Falcon-LLM Team. 2024. The falcon 3 family of open models.

Simone Tedeschi, Johan Bos, Thierry Declerck, Jan Hajič, Daniel Hershcovich, Eduard Hovy, Alexander Koller, Simon Krek, Steven Schockaert, Rico Sennrich, Ekaterina Shutova, and Roberto Navigli. 2023. What's the meaning of superhuman performance in today's NLU? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12471–12491, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Son Quoc Tran, Phong Nguyen-Thuan Do, Uyen Le, and Matt Kretchmar. 2023. The impacts of unanswerable questions on the robustness of machine reading comprehension models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1543–1557, Dubrovnik, Croatia. Association for Computational Linguistics.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022. Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yulong Wu, Viktor Schlegel, and Riza Batista-Navarro. 2021. Is the understanding of explicit discourse relations required in machine reading comprehension? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3565–3579, Online. Association for Computational Linguistics.

Yulong Wu, Viktor Schlegel, and Riza Batista-Navarro. 2023a. Are machine reading comprehension systems robust to context paraphrasing? In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 184–196, Nusa Dua, Bali. Association for Computational Linguistics.

Yulong Wu, Viktor Schlegel, and Riza Batista-Navarro. 2025. Natural context drift undermines the natural language understanding of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1248–1259, Suzhou, China. Association for Computational Linguistics.

Yulong Wu, Viktor Schlegel, Daniel Beck, and Riza Batista-Navarro. 2023b. MMT's submission for the WMT 2023 quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 856–862, Singapore. Association for Computational Linguistics.

Jun Yan, Yang Xiao, Sagnik Mukherjee, Bill Yuchen Lin, Robin Jia, and Xiang Ren. 2022. On the robustness of reading comprehension models to entity renaming. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 508–520, Seattle, United States. Association for Computational Linguistics.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in Wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2000–2010, Copenhagen, Denmark. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Kun Zhang, Le Wu, Kui Yu, Guangyi Lv, and Dacao Zhang. 2025. Evaluating and improving robustness in large language models: A survey and future directions. *Preprint*, arXiv:2506.11111.

A Encoder-only Model Parameters and Hyperparameters for Fine-tuning

Table 7 shows the hyperparameters used to fine-tune the pre-trained encoder-only QA models in this work and their number of parameters contained.

Model$_{Parameters(M)}$	d	b	lr	ep
DistilBERT$_{(66)}$	384	8	$3e-5$	3
BERT$_{(110/340)}$	384	8	$3e-5$	2
SpanBERT$_{(110/340)}$	512	4	$2e-5$	4
RoBERTa$_{(125/355)}$	384	8	$3e-5$	2
ALBERT$_{(11/17/223)}$	384	4	$3e-5$	2
DeBERTa$_{(350)}$	384	4	$3e-6$	3

Table 7: Number of parameters in each type of pre-trained encoder-only QA model and the hyperparameters used to fine-tune them. For BERT, SpanBERT, RoBERTa and ALBERT, we show the number of model parameters in the order of *base*, *large* and *xxlarge* (if applicable) version. d is the size of the token sequence fed into the model, b is the training batch size, lr is the learning rate, and ep is the number of training epochs. We used stride = 128 for documents longer than d tokens.

B Instruction Templates for Flan-T5 Evaluation

In Table 8, we present the instruction templates employed in constructing the inputs to the Flan-T5 model for the SQuAD 1.1 format and SQuAD 2.0 format test sets, respectively.

C QA Prompts

We use the following zero-shot prompts to instruct the decoder-only models to generate responses in the task of QA.

SQuAD 1.1 & TyDi QA: *Use the provided article delimited by triple quotes to answer question. Provide only the shortest continuous span from the context without any additional explanation.*\n\n"""*{context}*"""\n\n*Question: {question}*

	SQuAD 1.1
1	“Read this and answer the question\n\n{context}\n\n{question}”
2	“{context}\n{question}”
3	“Answer a question about this article:\n{context}\n{question}”
4	“Here is a question about this article: {context}\nWhat is the answer to this question: {question}”
5	“Article: {context}\n\nQuestion: {question}”
6	“Article: {context}\n\nNow answer this question: {question}”
	SQuAD 2.0
1	“Read this and answer the question. If the question is unanswerable, say \“unanswerable\".\n\n{context}\n\n{question}”
2	“{context}\n{question} (If the question is unanswerable, say \“unanswerable\")”
3	“{context}\nTry to answer this question if possible (otherwise reply \“unanswerable\"): {question}”
4	“{context}\nIf it is possible to answer this question, answer it for me (else, reply \“unanswerable\"): {question}”
5	“{context}\n\nAnswer this question, if possible (if impossible, reply \“unanswerable\"): {question}”
6	“Read this: {context}\nNow answer this question, if there is an answer (If it cannot be answered, return \“unanswerable\"): {question}”

Table 8: Various instruction templates for `Flan-T5` model evaluation.

SQuAD 2.0: *Use the provided article delimited by triple quotes to answer question. Provide only the shortest continuous span from the context without any additional explanation. If the question is unanswerable, return “unanswerable".*\n\n*“““{context}”””*\n\n*Question: {question}*

DROP & HotpotQA: *Use the provided article delimited by triple quotes to answer question. Provide only the answer without any additional explanation.*\n\n*“““{context}”””*\n\n*Question: {question}*

BoolQ: *Use the provided article delimited by triple quotes to answer question. Return only TRUE or FALSE.*\n\n*“““{context}”””*\n\n*Question: {question}*

D Indicators of Unanswerable

We manually identify a set of phrases contained in the output of LLMs that indicate the unanswerability of the question, including *“I cannot answer this/the question”, “unanswerable”, “There is no indication in the provided article”, “The context provided does not provide enough information”, “There is no reference in the given article”, “The answer to the question is not provided in the given article”, “it is not possible”, “question cannot be answered”* and *“context/question/article/text/article provided/passage does not”*.

E Human Annotation Instructions

In Figure 5, we show the instructions given to human annotators for error analysis and adversarial validity checking, respectively. All our human annotators are university students in the United Kingdom and China. Before commencing each task, we ask the annotators to annotate some examples and report the average time spent on each. As compensation, annotators receive 40 pence for each annotated example.

F Demonstration of Perturbed QA Examples for Encoder-only Models

Figure 6 illustrates a naturally perturbed QA instance each for categories C2W, C2C, and W2C, with the annotated perturbation type(s).

G Process of Adversarial Validity Verification

We first present two human annotators with the same collection of adversarial instances, which includes only perturbed contexts and their corresponding questions, and then ask them to answer the question based on the perturbed context. The annotators are required to select the shortest continuous span in the perturbed context that answers the question and are allowed to leave the answer blank if they are confident that the question is not answerable. Full instructions given to the annotators can be seen in Appendix E. Subsequently, for both annotators, we measure the correctness (1 or 0) of their provided answers by comparing each of them with the corresponding ground truth answers[8]. The inter-annotator agreement is then

[8]Here, as long as one of the ground truth answers is included in the human-provided answer span, we consider the

Error Analysis

You will be presented with pairs of reading contexts and their modified versions. The task is to compare each context and its modified version, observe the changes made and classify them into one or more of the semantic edit intention categories detailed below:

- *Copy Editing*: Rephrase; improve grammar, spelling, tone, or punctuation
- *Clarification*: Specify or explain an existing fact or meaning by example or discussion without adding new information
- *Elaboration*: Extend/add new content; insert a fact or new meaningful assertion
- *Fact Update*: Update numbers, dates, scores, episodes, status, etc. based on newly available information
- *Refactoring*: Restructure the article; move and rewrite content, without changing the meaning of it
- *Simplification*: Reduce the complexity or breadth of discussion; may remove information
- *Vandalism*: Deliberately attempt to damage the article
- *Other*: None of the above

We will use your annotation to calculate the percentage of each edit category.

Adversarial Validity Checking

Please read each provided context carefully and answer a corresponding question. Select the shortest continuous span from the context as your answer. If you believe a question cannot be answered, leave the answer blank. Your answer will be compared with the ground truth answers, and the result will only be used to decide the human answerability of the question.

Figure 5: Instructions for the two distinct human annotation tasks. In the error analysis task, the eight semantic edit intentions are adopted from (Yang et al., 2017).

measured by computing the Cohen's κ coefficient (Cohen, 1960). We then involve a third human annotator to annotate the adversarial examples on which the first two annotators disagree and then take the majority label as ground truth.

H QA Under Synthetic Perturbations

To examine differences between natural and synthetic perturbations, we evaluate performance changes across QA model architectures under varying levels of synthetic noise (Table 9). While certain character-level and word-level perturbation methods have been investigated across multiple NLP tasks, such as the CharSwapMid for machine translation (Belinkov and Bisk, 2018) and the WDelete for quality estimation (Kanojia et al., 2021; Wu et al., 2023b), none of these has been applied to the contextual paragraph to study the robustness for the task of QA. Results are shown in Table 10. For each method, from its created SQuAD 2.0-format test set pair, we also randomly select 50 instances where the `GPT-3.5-turbo-0125` shows evidence of being not robust, resulting in a total of 800 adversarial examples. We then measure their validity using the methodology described in Appendix G, shuffling their order to mitigate potential bias, and present the results (0.81 Cohen's κ) in Table 11.

Table 10 shows that QA systems generally exhibit limited robustness to synthetic perturbations, with varying degrees of performance degradation. Overall, there is no substantial difference between natural and synthetic perturbations in inducing model failures, although methods such as AddSent, WSplit and WInsert (WE) produce more pronounced performance drops. However, the reliability of these drops is questionable due to their low human-verified validity scores, whereas natural perturbations generally achieve higher validity than synthetic ones.

I Exhaustive Search Algorithm for Challenging Test Set Construction

We propose an exhaustive search algorithm that leverages the predictions of all encoder-only models to create the challenging natural perturbed test set. In detailed terms, for each matched reading passage from the prior version and its counterpart from the current version, we determine which should be designated as the *original* and which as the *perturbed* based on which scenario can yield the questions on which the maximum sum of the number

prediction to be correct.

Category: C2W

Original Paragraph: *Jacksonville, like most large cities in the United States, suffered from negative effects of rapid urban sprawl after World War II. The construction of highways led residents to move to newer housing in the suburbs. After World War II, the government of the city of Jacksonville began to increase spending to fund new public building projects in the boom that occurred after the war.* [. . .]
Perturbed Paragraph: *Jacksonville, like most large cities in the United States, suffered from negative effects of rapid urban sprawl after World War V. The construction of highways led residents to move to newer housing in the suburbs. After World War II, the government of the city of Jacksonville began to increase spending to fund new public building projects in the boom that occurred after the war.* [. . .]
Question: What did Jacksonville suffer from following World War I?
Prediction of `distilbert-base` and `spanbert-large-cased`: unanswerable→rapid urban sprawl
Annotated Natural Perturbation Type: Vandalism

Category: C2C

Original Paragraph: *Construction projects can suffer from preventable financial problems. Underbids happen when builders ask for too little money to complete the project. Cash flow problems exist when the present amount of funding cannot cover the current costs for labour and materials, and because they are a matter of having sufficient funds at a specific time, can arise even when the overall total is enough. Fraud is a problem in many fields, but is notoriously prevalent in the construction field. Financial planning for the project is intended to ensure that a solid plan with adequate safeguards and contingency plans are in place before the project is started and is required to ensure that the plan is properly executed over the life of the project.*
Perturbed Paragraph: *Financial planning ensures adequate safeguards and contingency plans are in place before the project is started, and ensures that the plan is properly executed over the life of the project. Construction projects can suffer from preventable financial problems. Underbids happen when builders ask for too little money to complete the project. Cash flow problems exist when the present amount of funding cannot cover the current costs for labour and materials; such problems may arise even when the overall budget is adequate, presenting a temporary issue. Fraud is also an occasional construction issue.*
Question: What can construction projects suffer from?
Prediction of all encoder-only models: preventable financial problems→preventable financial problems
Annotated Natural Perturbation Type: Copy Editing; Refactoring; Simplification

Category: W2C

Original Paragraph: *[. . .] The antigens expressed by tumors have several sources; some are derived from oncogenic viruses like human papillomavirus, which causes cervical cancer, while others are the organism's own proteins that occur at low levels in normal cells but reach high levels in tumor cells. [. . .] A third possible source of tumor antigens are proteins normally important for regulating cell growth and survival, that commonly mutate into cancer inducing molecules called oncogenes.*
Perturbed Paragraph: *[. . .] The antigens expressed by tumors have several sources; some are derived from oncogenic viruses like human papillomavirus, which causes cancer of the cervix, vulva, vagina, penis, anus, mouth, and throat,while others are the organism's own proteins that occur at low levels in normal cells but reach high levels in tumor cells. [. . .] A third possible source of tumor antigens are proteins normally important for regulating cell growth and survival, that commonly mutate into cancer inducing molecules called oncogenes.*
Question: What is a fourth possible source for tumor antigens?
Prediction of `bert-base-uncased`: proteins normally important for regulating cell growth and survival→unanswerable
Annotated Natural Perturbation Type: Elaboration

Figure 6: Natural perturbed QA example in C2W, C2C and W2C categories.

Method	Explanation
	character-level
CharOCR	Replace characters with predefined Optical Character Recognition (OCR) errors.
CharInsert	Inject new characters randomly.
CharSubstitute	Substitute original characters randomly.
CharSwapMid	Swap adjacent characters within words randomly, excluding the first and last character.
CharSwapRand	Swap characters randomly without constraint.
	word-level
WInsert (CWE)	Insert new words to random position according to contextual word embeddings calculation from RoBERTa-base.
WSubstitute (CWE)	Substitute words according to contextual word embeddings calculation from RoBERTa-base (Liu et al., 2019).
WSplit	Split words to two tokens randomly.
WSwap	Swap adjacent words randomly.
WDelete	Delete words randomly.
WCrop	Remove a set of continuous word randomly.
Word Synonym Substitution (WSynSub)	Substitute words with synonyms from large size English PPDB (Pavlick et al., 2015).
WInsert (WE)	Insert new words to random position according to GloVe (Pennington et al., 2014) word embeddings calculation[9].
	sentence-level
AddSent	Add a context-irrelevant distractor sentence with high lexical overlap to the question at the beginning of the context.
	document-level
Document Paraphrasing (DocPara)	Paraphrasing the whole context paragraph directly.
Style Transfer	Rephrase the passage using a distinct persona discerned based on its topic.

Table 9: Various synthetic perturbation approaches.

of encoder-only models demonstrates the lack of robustness phenomenon[10]. To be specific:

Given a matched reading passage (P) from the prior version, its counterpart (P') from the current version, and the associated questions:

First Scenario: We treat (P) as the original passage and (P') as the perturbed one. We then evaluate, for each associated question, how many encoder-only models demonstrate the lack of robustness phenomenon, i.e., succeed on (P) but fail on (P'). We finally obtain the total number of models that demonstrate the lack of robustness phenomenon across all questions, denoted as (N). Questions on which none of the models demonstrate the lack of robustness phenomenon are removed, leaving (Q) questions.

Second Scenario: We treat (P') as the original passage and (P) as the perturbed one. We then repeat the same evaluation process as described in the first scenario and obtain the total number of models demonstrating the lack of robustness phenomenon across all questions, denoted as (N'). Questions on which none of the models demonstrate the lack of robustness phenomenon are removed as well, leaving (Q') questions.

If ($N > N'$), we consider (P) as the original passage and (P') as the perturbed version.

If ($N < N'$), we consider (P') as the original and (P) as the perturbed.

If ($N = N'$), we compare (Q) and (Q'):

- If ($Q > Q'$), we consider (P) as the original passage and (P') as the perturbed version.
- If ($Q < Q'$), we consider (P') as the original and (P) as the perturbed.
- If ($Q = Q'$), the order does not matter, and we randomly decide which one should be the original and which should be the perturbed.

We finally process the identified original and perturbed passage pairs to ensure that the original passages are within the original SQuAD 1.1 development set. For those original passages with multiple occurrences, we select the one with the maximum number of questions reserved.

J Natural Adversarial Samples for LLMs

We demonstrate two naturally perturbed QA examples that pose challenges for LLMs in Figure 7.

K Impact of Synthetic Adversarial Training

Figure 8 describes the impact of synthetic adversarial training (for `deberta-large`) on handling natural and synthetic perturbations.

[10]We define A model as lacking robustness to the perturbation if it achieves 1 EM on the original question but attains less than 0.4 F1 on the perturbed one (for answerable questions).

NAT_V1_CHALLENGE

Original Paragraph: *In business, notable alumni include Microsoft CEO Satya Nadella, Oracle Corporation founder and the third richest man in America Larry Ellison, Goldman Sachs and MF Global CEO as well as former Governor of New Jersey Jon Corzine, McKinsey & Company founder and author of the first management accounting textbook James O. McKinsey, Arley D. Cathey, Bloomberg L.P. CEO Daniel Doctoroff, Credit Suisse CEO Brady Dougan, Morningstar, Inc. founder and CEO Joe Mansueto, Chicago Cubs owner and chairman Thomas S. Ricketts, and NBA commissioner Adam Silver.*

Perturbed Paragraph: *In business, notable alumni include Microsoft CEO Satya Nadella, Oracle Corporation founder and the third richest man in America Larry Ellison, Goldman Sachs and MF Global CEO as well as former Governor of New Jersey Jon Corzine, McKinsey & Company founder and author of the first management accounting textbook James O. McKinsey, co-founder of the Blackstone Group Peter G. Peterson, co-founder of AQR Capital Management Cliff Asness, founder of Dimensional Fund Advisors David Booth, founder of The Carlyle Group David Rubenstein, Lazard CEO Ken Jacobs, entrepreneur David O. Sacks, CEO of TPG Group and former COO of Goldman Sachs Jon Winkelreid, former COO of Goldman Sachs Andrew Alper, billionaire investor and founder of Oaktree Capital Management Howard Marks, Bloomberg L.P. CEO Daniel Doctoroff, Credit Suisse CEO Brady Dougan, Morningstar, Inc. founder and CEO Joe Mansueto, Chicago Cubs owner and chairman Thomas S. Ricketts, and NBA commissioner Adam Silver.*

Question: What Goldman Sachs CEO is also an alumni of the University of Chicago?

Prediction of `GPT-3.5-turbo-0125` and `Llama-3-8B-Instruct`: Jon Corzine→Jon Winkelreid

Prediction of `Falcon-40B-Instruct`: Jon Corzine→David Rubenstein, co-founder of The Carlyle Group, is also an alumnus of the University of Chicago.

NAT_V2_CHALLENGE

Original Paragraph: *Each chapter has a number of authors who are responsible for writing and editing the material. A chapter typically has two "coordinating lead authors", ten to fifteen "lead authors", and a somewhat larger number of "contributing authors". The coordinating lead authors are responsible for assembling the contributions of the other authors, ensuring that they meet stylistic and formatting requirements, and reporting to the Working Group chairs. Lead authors are responsible for writing sections of chapters. Contributing authors prepare text, graphs or data for inclusion by the lead authors.*

Perturbed Paragraph: *Each chapter has a number of authors to write and edit the material. A typical chapter has two coordinating lead authors, ten to fifteen lead authors and a larger number of contributing authors. The coordinating lead authors assemble the contributions of the other authors. They ensure that contributions meet stylistic and formatting requirements. They report to the Working Group co-chairs. Lead authors write sections of chapters. They invite contributing authors to prepare text, graphs or data for inclusion.*

Question: Who has the responsibility for publishing materials?

Prediction of `Mistral-7B-Instruct-v0.2`: Unanswerable. The text does not mention any responsibility related to publishing materials.→The coordinating lead authors are responsible for publishing materials in the given context.

Figure 7: Natural perturbed QA examples that confuse LLMs.

Dataset	Victim \ Attacker	CharOCR	CharInsert	CharSubstitute	CharSwapMid	CharSwapRand	WInsert (CWE)	WSubstitute (CWE)	WSplit	WSwap	WDelete	WCrop	WSynSub	WInsert (WE)	AddSent	DocPara	Style Transfer
SQuAD 1.1	distilbert-base	-5.72	-8.71	-8.27	-8.18	-8.28	-2.27	-3.59	-28.25	-2.05	-2.17	-1.33	-9.78	-25.93	-14.69	-6.79	-7.28
	bert-base-cased	-4.14	-6.6	-6.82	-6.04	-5.95	-2.22	-3.21	-21.57	-2.05	-2.29	-1.32	-8.67	-18.26	-22.95	-7.0	-6.59
	bert-base-uncased	-4.65	-7.85	-7.32	-7.3	-6.91	-2.18	-3.31	-26.02	-1.83	-2.2	-1.44	-9.25	-23.99	-24.08	-6.15	-6.63
	bert-large-cased	-2.04	-3.84	-4.01	-3.96	-4.2	-2.18	-2.94	-14.19	-1.38	-1.54	-1.32	-6.71	-12.67	-26.08	-6.7	-5.55
	bert-large-uncased	-2.78	-4.34	-5.11	-5.06	-4.53	-2.21	-3.34	-20.55	-1.16	-1.88	-1.42	-8.12	-18.55	-22.19	-5.82	-5.61
	spanbert-base-cased	-3.77	-6.48	-7.31	-7.06	-7.02	-2.45	-3.46	-24.71	-2.21	-2.34	-1.68	-8.84	-21.36	-17.3	-6.37	-5.73
	spanbert-large-cased	-2.7	-5.42	-5.66	-5.54	-5.48	-2.14	-3.19	-22.36	-1.65	-2.37	-1.51	-8.82	-18.32	-16.24	-6.41	-5.64
	roberta-base	-2.26	-3.77	-4.72	-4.19	-3.99	-1.88	-2.97	-12.19	-1.08	-1.84	-1.27	-6.11	-12.18	-13.98	-6.45	-5.42
	roberta-large	-1.57	-2.34	-2.5	-2.83	-3.08	-1.8	-2.83	-9.11	-1.09	-1.44	-0.93	-5.11	-9.01	-14.19	-5.65	-4.75
	albert-base-v1	-4.94	-7.54	-8.11	-7.59	-7.6	-2.52	-3.72	-27.79	-2.02	-2.61	-1.56	-8.65	-23.01	-25.92	-6.73	-6.92
	albert-base-v2	-3.79	-5.65	-6.47	-6.79	-6.24	-2.1	-3.46	-23.17	-1.87	-2.33	-1.59	-8.59	-18.75	-23.59	-7.19	-6.18
	albert-large-v1	-3.55	-5.22	-5.89	-6.27	-6.03	-2.08	-3.21	-20.56	-1.92	-2.23	-1.54	-8.75	-18.73	-19.52	-7.37	-6.47
	albert-large-v2	-3.33	-5.07	-5.31	-5.81	-6.05	-1.94	-3.1	-21.54	-1.86	-2.8	-1.39	-8.21	-18.76	-19.52	-6.96	-6.14
	albert-xxlarge-v1	-1.43	-2.06	-2.44	-2.82	-2.68	-1.7	-2.95	-10.4	-1.12	-1.84	-1.15	-4.94	-7.61	-15.31	-5.59	-4.68
	albert-xxlarge-v2	-1.28	-2.08	-2.44	-2.98	-2.77	-1.76	-2.75	-11.68	-1.11	-1.5	-1.27	-5.11	-9.03	-13.19	-5.24	-4.64
	deberta-large	-0.86	-1.54	-1.63	-1.78	-2.24	-1.36	-2.68	-7.23	-0.51	-1.09	-1.06	-3.95	-4.83	-11.52	-4.98	-4.45
	flan-t5-small	-5.31	-7.93	-8.83	-7.34	-7.49	-2.16	-5.02	-20.88	-2.95	-2.88	-1.49	-10.02	-19.66	-12.51	-7.85	-7.96
	flan-t5-base	-2.34	-3.32	-4.54	-4.36	-4.8	-2.06	-4.33	-11.95	-1.45	-2.7	-1.71	-8.29	-9.67	-11.54	-7.05	-6.03
	flan-t5-large	-1.16	-1.24	-2.01	-1.5	-2.15	-1.98	-4.79	-6.43	-0.72	-1.61	-1.29	-5.35	-4.08	-9.3	-5.97	-5.18
	flan-t5-xl	-1.14	-1.22	-1.94	-1.94	-2.23	-1.67	-4.44	-5.43	-1.12	-1.13	-1.06	-3.83	-1.92	-8.05	-5.35	-4.65
	GPT-3.5-turbo-0125	-0.14	-0.34	-0.59	-0.88	-0.69	-0.89	-1.74	0.17	-0.9	-1.0	-0.7	-4.14	-1.02	-9.22	-3.57	-3.82
	Gemma-2B-IT	-3.5	-5.46	-12.18	-6.53	-7.9	-3.62	-11.93	-12.47	-3.3	-8.88	-4.53	-39.87	-18.93	-4.57	-15.46	-14.99
	Gemma-7B-IT	-2.17	-2.59	-5.28	-3.96	-4.41	-1.38	-6.74	-3.69	-1.76	-5.36	-3.5	-14.97	-5.97	-11.43	-11.6	-10.25
	Llama 2-chat-7B	-0.82	-0.46	-1.36	-1.59	-1.89	-0.86	-3.16	-1.06	-0.83	-1.25	-0.64	-5.15	-2.94	-11.79	-4.5	-3.76
	Llama 2-chat-13B	-1.0	-0.7	-1.17	-1.35	-1.34	-0.61	-3.2	-0.51	-1.1	-1.54	-1.87	-4.61	-1.68	-6.97	-5.41	-4.21
	Llama-3-8B-Instruct	-0.49	-0.73	-1.52	-1.53	-1.43	-0.34	-3.18	-0.64	-0.99	-1.23	-1.1	-4.42	-1.43	-10.79	-5.5	-4.37
	Mistral-7B-Instruct-v0.2	-0.97	-0.42	-1.36	-0.96	-1.27	-0.31	-2.02	-1.46	-1.1	-1.77	-1.13	-4.75	-1.94	-5.25	-4.42	-3.83
	Falcon-7B-Instruct	-3.81	-6.25	-6.68	-6.43	-4.9	-2.73	-3.19	-9.37	-3.87	-3.61	-2.41	-12.98	-9.95	-8.01	-4.89	-2.75
	Falcon-40B-Instruct	-1.83	-0.73	-3.56	-2.37	-2.68	-1.77	-4.21	-0.88	-0.9	-2.61	-1.65	-6.91	-3.3	-14.66	-6.32	-4.32
SQuAD 2.0	distilbert-base	-2.65(−9.96/5.42)	-3.96(−19.14/12.57)	-5.38(−19.6/10.27)	-4.28(−18.33/10.99)	-5.16(−18.12/9.23)	-1.4(−4.41/1.98)	-2.63(−9.65/5.23)	-11.56(−59.55/35.71)	-0.64(−2.3/1.17)	-1.64(−6.0/3.18)	-1.36(−3.38/0.9)	-6.59(−23.67/12.97)	-10.56(−43.28/22.85)	-24.02(−21.13/−27.36)	-5.58(−16.31/6.58)	-4.28(−14.6/7.31)
	bert-base-cased	-3.73(−10.33/2.83)	-5.88(−17.58/5.53)	-5.47(−17.0/5.95)	-5.86(−16.27/4.36)	-5.84(−16.02/4.31)	-1.48(−3.68/0.74)	-3.88(−9.6/1.85)	-13.54(−47.25/15.11)	-0.91(−1.88/0.07)	-2.26(−6.11/1.54)	-1.52(−3.81/0.79)	-8.53(−21.69/4.83)	-9.72(−36.21/14.6)	-31.3(−37.28/−25.26)	-6.77(−16.31/2.85)	-6.9(−14.93/1.14)
	bert-base-uncased	-3.82(−11.45/3.91)	-6.02(−17.34/5.11)	-5.83(−17.29/5.64)	-5.27(−17.17/6.54)	-5.83(−16.85/5.26)	-1.03(−3.97/1.96)	-4.13(−11.19/3.03)	-15.7(−55.42/19.32)	-0.43(−1.59/0.73)	-2.4(−6.43/1.63)	-1.71(−4.43/1.05)	-9.42(−23.32/4.86)	-12.94(−36.69/9.24)	-33.02(−35.11/−30.86)	-6.45(−16.24/3.48)	-5.94(−14.64/2.78)
	bert-large-cased	-2.47(−6.61/1.65)	-4.86(−12.68/2.81)	-4.92(−12.64/2.78)	-5.52(−12.95/1.78)	-5.07(−12.15/2.01)	-1.2(−2.9/0.51)	-4.92(−11.17/1.35)	-15.1(−37.91/4.79)	-0.26(−1.15/0.61)	-3.21(−7.19/0.76)	-1.88(−4.28/0.56)	-10.34(−19.71/−0.91)	-6.77(−19.78/5.23)	-34.27(−46.81/−21.27)	-8.1(−15.52/−0.67)	-7.21(−13.01/−1.48)
	bert-large-uncased	-3.9(−8.21/0.36)	-5.34(−12.99/2.12)	-6.11(−13.99/1.69)	-5.61(−12.75/1.43)	-5.46(−12.48/1.5)	-1.66(−3.46/0.1)	-5.0(−11.07/1.02)	-16.73(−48.06/10.16)	-0.82(−2.08/0.4)	-2.75(−6.23/0.68)	-1.63(−3.72/0.48)	-10.98(−21.36/−0.51)	-11.27(−27.53/3.73)	-35.79(−44.61/−26.73)	-7.86(−15.76/0.0)	-7.54(−14.6/−0.57)
	spanbert-base-cased	-4.43(−8.05/−0.89)	-7.51(−14.59/−0.75)	-6.83(−14.95/0.96)	-7.11(−14.32/−0.2)	-7.84(−15.11/−0.86)	-1.74(−3.47/−0.05)	-5.56(−12.17/0.83)	-21.04(−39.6/−5.38)	-2.34(−2.16/−2.5)	-3.93(−6.44/−1.55)	-2.3(−3.74/−0.9)	-11.49(−17.9/−5.3)	-14.7(−26.02/−4.66)	-28.62(−32.9/−24.3)	-8.7(−15.36/−2.27)	-7.8(−15.05/−0.88)
	spanbert-large-cased	-3.91(−7.29/−0.74)	-7.08(−12.91/−1.67)	-6.83(−13.09/−1.0)	-7.32(−12.82/−2.21)	-7.46(−14.24/−1.11)	-1.49(−2.93/−0.17)	-5.7(−11.39/−0.35)	-18.52(−33.89/−6.04)	-2.14(−2.43/−1.86)	-3.4(−5.48/−1.46)	-2.02(−3.39/−0.71)	-11.39(−16.92/−6.24)	-13.11(−24.15/−3.59)	-29.65(−38.44/−21.03)	-8.6(−15.2/−2.46)	-7.42(−12.67/−2.55)
	roberta-base	-3.05(−5.9/−0.29)	-5.45(−11.83/0.55)	-5.37(−12.15/1.11)	-5.52(−11.85/0.44)	-5.53(−12.68/1.29)	-1.22(−3.1/0.58)	-4.97(−11.87/1.64)	-15.02(−30.86/−1.85)	-0.99(−1.81/−0.21)	-3.56(−7.25/−0.06)	-2.15(−3.88/−0.46)	-8.84(−15.77/−2.2)	-8.94(−19.32/0.23)	-25.73(−32.6/−18.85)	-8.05(−15.67/−0.78)	-7.69(−13.92/−1.8)
	roberta-large	-2.54(−4.11/−1.09)	-4.52(−7.69/−1.64)	-4.92(−8.19/−1.9)	-4.92(−7.76/−2.32)	-4.91(−8.44/−1.65)	-1.71(−3.41/−0.12)	-6.04(−12.93/0.36)	-11.91(−16.9/−7.92)	-1.55(−2.04/−1.1)	-3.5(−5.72/−1.44)	-2.28(−3.97/−0.67)	-10.58(−15.72/−5.83)	-8.69(−14.3/−3.9)	-25.66(−32.82/−18.52)	-8.33(−15.61/−1.63)	-7.37(−13.61/−1.63)
	albert-base-v1	-5.62(−12.18/0.23)	-8.7(−20.33/1.45)	-7.33(−19.14/3.14)	-8.17(−19.48/1.75)	-7.76(−18.59/1.98)	-1.76(−4.55/0.68)	-5.64(−12.99/0.9)	-20.95(−58.64/8.05)	-1.72(−4.08/0.34)	-4.62(−10.4/0.5)	-2.68(−5.56/−0.07)	-11.72(−23.83/−0.73)	-14.37(−33.25/1.12)	-36.85(−46.13/−28.13)	-9.64(−20.38/−0.09)	-8.71(−18.82/0.31)
	albert-base-v2	-5.17(−11.0/0.17)	-6.82(−16.17/1.51)	-7.91(−17.32/0.62)	-7.71(−16.9/0.55)	-7.14(−16.01/0.91)	-1.58(−4.05/0.65)	-5.84(−13.27/0.9)	-19.59(−52.41/6.15)	-2.0(−3.14/−0.99)	-4.09(−8.51/−0.08)	-2.61(−5.07/−0.38)	-10.99(−21.88/−0.99)	-11.7(−25.83/0.23)	-34.77(−41.78/−28.19)	-8.87(−17.83/−0.7)	-7.82(−16.47/0.02)
	albert-large-v1	-4.46(−10.28/0.93)	-7.91(−17.6/0.81)	-7.93(−18.27/1.46)	-7.69(−17.17/1.0)	-7.56(−17.03/1.13)	-1.67(−3.65/0.15)	-5.66(−13.24/1.36)	-19.28(−50.7/5.73)	-1.08(−3.08/0.75)	-3.75(−8.51/0.61)	-2.53(−5.72/0.43)	-12.02(−21.5/−3.11)	-13.36(−30.25/1.0)	-34.89(−40.81/−29.09)	-8.93(−17.81/−0.64)	-7.73(−16.31/0.21)
	albert-large-v2	-3.85(−8.34/0.38)	-6.45(−13.89/0.38)	-6.24(−13.93/1.01)	-6.34(−13.46/0.26)	-6.03(−12.75/0.26)	-1.46(−3.33/0.3)	-5.56(−12.35/0.88)	-19.77(−44.92/0.62)	-1.48(−2.72/−0.34)	-3.9(−8.15/0.06)	-2.69(−5.04/−0.45)	-11.43(−21.76/−1.6)	-10.57(−22.42/−0.27)	-31.77(−38.15/−25.49)	-8.91(−16.42/−1.88)	-8.2(−15.52/−1.37)
	albert-xxlarge-v1	-2.43(−2.92/−1.94)	-4.19(−5.59/−2.9)	-4.31(−5.05/−3.6)	-4.18(−5.77/−2.67)	-4.51(−7.14/−2.03)	-1.49(−2.64/−0.39)	-5.99(−11.76/−0.53)	-13.22(−18.41/−8.95)	-1.47(−1.62/−1.33)	-3.55(−5.68/−1.54)	-2.3(−4.02/−0.65)	-9.04(−12.65/−5.66)	-7.13(−12.51/−2.44)	-26.12(−28.58/−23.67)	-8.02(−14.92/−1.54)	-7.13(−12.9/−1.77)
	albert-xxlarge-v2	-2.25(−2.61/−1.9)	-3.91(−5.29/−2.59)	-4.51(−6.0/−3.09)	-5.05(−6.89/−3.32)	-4.86(−7.59/−2.25)	-1.46(−2.64/−0.35)	-5.68(−11.2/−0.39)	-13.52(−21.09/−7.21)	-1.3(−1.72/−0.91)	-3.23(−5.13/−1.44)	-1.96(−3.58/−0.4)	-8.8(−11.95/−5.82)	-7.33(−11.83/−3.37)	-25.06(−29.04/−21.07)	-8.41(−14.41/−2.73)	-6.92(−12.18/−1.98)
	deberta-large	-1.39(−1.62/−1.17)	-2.94(−2.37/−3.46)	-3.05(−3.5/−2.63)	-2.89(−3.19/−2.64)	-3.24(−4.18/−2.37)	-1.32(−2.6/−0.12)	-5.93(−12.1/−0.18)	-9.31(−10.38/−8.44)	-1.52(−1.6/−1.46)	-2.87(−4.39/−1.48)	-1.89(−3.37/−0.47)	-8.34(−9.08/−7.66)	-5.95(−5.89/−6.02)	-22.96(−27.86/−18.19)	-7.35(−13.84/−1.32)	-7.43(−12.99/−2.29)
	flan-t5-small	-2.07(−7.91/5.18)	-2.52(−10.98/7.92)	-2.37(−12.21/9.93)	-3.09(−11.09/6.77)	-2.28(−10.17/7.54)	-0.73(−3.73/3.05)	-2.05(−7.75/5.24)	-8.7(−31.68/15.87)	-0.12(−3.07/3.58)	-0.71(−5.68/5.5)	-1.27(−4.19/2.38)	-3.73(−24.18/23.62)	-7.04(−29.75/19.74)	-15.79(−15.74/−15.82)	-3.08(−17.91/16.18)	-2.88(−14.63/12.24)
	flan-t5-base	-2.35(−5.22/0.77)	-4.75(−9.59/0.36)	-4.85(−10.24/0.95)	-4.22(−9.03/0.9)	-4.77(−10.68/1.59)	-1.35(−3.27/0.73)	-4.34(−11.01/2.96)	-12.36(−21.26/−3.86)	-0.79(−1.65/0.11)	-3.22(−6.73/0.56)	-1.94(−4.19/0.51)	-8.01(−15.53/−0.4)	-25.97(−22.91/−29.41)	-7.63(−14.9/0.28)	-6.98(−11.69/−1.89)	
	flan-t5-large	-2.23(−2.34/−2.15)	-4.09(−3.84/−4.32)	-4.07(−5.04/−3.15)	-3.69(−4.19/−3.21)	-4.24(−5.35/−3.18)	-1.7(−3.05/−0.42)	-5.95(−12.45/0.35)	-11.41(−8.14/−14.15)	-2.12(−1.03/−3.14)	-3.84(−5.83/−1.93)	-2.2(−3.16/−1.28)	-9.24(−11.81/−6.78)	-7.43(−5.96/−9.01)	-20.67(−20.33/−21.03)	-7.38(−14.85/−0.26)	-6.37(−12.06/−0.99)
	flan-t5-xl	-1.16(−1.68/−0.68)	-2.65(−3.48/−1.92)	-2.96(−4.18/−1.86)	-2.9(−4.26/−1.67)	-3.12(−4.99/−1.45)	-1.62(−3.32/−0.09)	-6.07(−12.88/0.08)	-6.12(−6.35/−5.95)	-1.49(−1.52/−1.48)	-3.06(−4.99/−1.32)	-2.33(−4.21/−0.61)	-7.06(−9.32/−5.03)	-3.19(−2.39/−3.87)	-19.91(−23.35/−16.64)	-7.54(−15.67/−0.3)	-7.08(−13.87/−1.07)
	GPT-3.5-turbo-0125	-1.31(−0.62/−4.66)	-3.6(−0.86/−15.8)	-2.81(−0.79/−12.15)	-3.34(−0.89/−14.7)	-3.53(−1.4/−13.76)	-1.66(−0.52/−6.92)	-1.57(−2.69/3.8)	-1.52(−1.07/−3.2)	-1.8(−0.9/−5.96)	-1.7(−1.06/−4.77)	-1.2(−0.78/−3.16)	-7.6(−5.63/−15.55)	-5.15(−1.51/−19.98)	-11.98(−13.27/−6.33)	-0.99(−4.99/16.57)	-1.52(−5.74/17.73)
	Gemma-2B-IT	-2.42(−4.44/−1.12)	-2.99(−9.18/0.78)	-3.44(−10.72/1.17)	-3.14(−7.72/−0.36)	-2.09(−6.26/0.56)	-1.72(−2.7/−1.1)	-2.82(−13.06/3.74)	-5.47(−20.84/2.2)	-1.31(−2.61/−0.52)	-1.59(−8.83/2.98)	-1.43(−4.59/0.7)	-8.54(−45.35/12.48)	-7.89(−19.54/−1.78)	-7.13(1.34/−13.83)	-1.29(−18.04/9.5)	-1.75(−16.17/7.45)
	Gemma-7B-IT	-2.64(−7.04/1.99)	-2.9(−9.48/3.81)	-3.88(−11.07/3.6)	-4.0(−9.92/2.06)	-3.6(−10.24/3.33)	-1.73(−3.33/−0.06)	-3.2(−9.16/2.95)	-2.2(−8.95/3.57)	-3.14(−6.29/0.03)	-1.7(−8.68/5.59)	-1.19(−4.77/2.7)	-8.15(−26.33/9.48)	-4.07(−12.84/3.95)	-12.09(−15.06/−8.66)	-2.79(−16.59/11.26)	-2.54(−15.18/10.3)
	Llama 2-chat-7B	-0.49(0.12/−8.16)	-2.75(−0.84/−25.26)	-2.86(−1.37/−21.38)	-2.34(−0.95/−19.33)	-2.72(−1.23/−21.9)	-0.94(−0.26/−8.97)	-3.56(−2.62/−15.04)	-3.26(−0.5/−28.52)	-0.96(−0.23/−10.13)	-1.68(−0.78/−12.77)	-1.07(−0.84/−3.86)	-6.49(−4.73/−28.48)	-6.11(−2.71/−44.26)	-8.72(−12.06/29.64)	-3.48(−5.16/16.82)	-3.2(−3.69/3.08)
	Llama 2-chat-13B	-2.76(−0.94/−9.44)	-3.32(−1.3/−10.15)	-4.63(−0.77/−17.78)	-3.58(−1.55/−10.68)	-4.3(−2.33/−11.23)	-1.02(−0.44/−3.11)	-3.83(−3.76/−4.08)	-2.23(−2.49/−1.47)	-1.82(−0.17/−7.52)	-3.53(−1.66/−10.08)	-1.41(−0.71/−4.02)	-10.46(−6.26/−23.67)	-5.81(−3.68/−12.88)	-6.92(−8.75/0.0)	-4.87(−5.11/−4.12)	-2.62(−4.05/2.29)
	Llama-3-8B-Instruct	0.05(−0.36/1.79)	-2.42(−1.46/−6.41)	-2.54(−1.82/−5.61)	-3.2(−2.27/−7.01)	-3.02(−2.21/−6.34)	-1.49(−0.57/−5.19)	-3.23(−3.57/−1.8)	-1.49(−1.31/−2.19)	-1.37(−0.69/−4.15)	-2.13(−1.85/−3.27)	-0.23(−0.9/2.54)	-10.32(−5.38/−28.89)	-8.19(−1.24/−33.72)	-15.5(−13.24/−25.18)	-1.62(−6.22/16.19)	-1.23(−5.3/14.4)
	Mistral-7B-Instruct-v0.2	-0.8(−0.97/−0.42)	-1.11(−1.21/−0.87)	-0.82(−1.08/−0.22)	-2.14(−1.54/−3.51)	-2.0(−2.16/−1.6)	-0.31(−0.49/0.14)	-2.27(−3.98/1.6)	-1.05(−1.25/−0.64)	-0.73(−0.5/−1.19)	-0.98(−2.28/1.88)	-0.28(−1.42/2.36)	-5.81(−6.1/−5.18)	-9.83(−1.56/−27.06)	-6.71(−10.44/1.69)	1.47(−6.43/19.04)	1.11(−4.88/14.35)
	Falcon-7B-Instruct	-1.6(−3.07/10.07)	-3.07(−3.99/3.32)	-5.59(−5.42/−6.66)	-5.95(−4.96/−13.24)	-5.05(−6.98/12.76)	1.03(0.09/8.61)	-5.41(−6.68/3.93)	-12.67(−13.56/−6.42)	-4.3(−4.32/−3.83)	-2.94(−3.42/0.74)	-2.01(−3.15/7.15)	-9.88(−12.32/13.25)	-7.89(−9.23/2.4)	-8.59(−11.81/11.38)	-5.31(−5.86/−1.36)	-0.29(−0.89/4.19)
	Falcon-40B-Instruct	-2.17(−2.26/−1.69)	-2.65(−2.33/−4.42)	-3.07(−2.62/−5.97)	-3.64(−3.09/−6.85)	-3.62(−2.98/−7.39)	-0.02(−0.28/1.39)	-4.06(−4.34/−2.37)	-1.47(0.0/−8.5)	-1.25(−1.07/−2.26)	-4.86(−6.4/4.84)	-1.66(−1.65/−1.78)	-8.54(−7.22/−16.2)	-5.53(−3.49/−16.99)	-10.53(−14.95/19.65)	-3.89(−6.03/8.24)	-5.28(−5.54/−3.9)

Table 10: Robustness of QA models under synthetic perturbations. For encoder-only and encoder-decoder models, the results displayed are the relative F1 change (%), while for decoder-only models, we demonstrate the relative IM change (%). In SQuAD 2.0, the values shown in the parentheses represent the relative change for answerable and unanswerable questions, respectively.

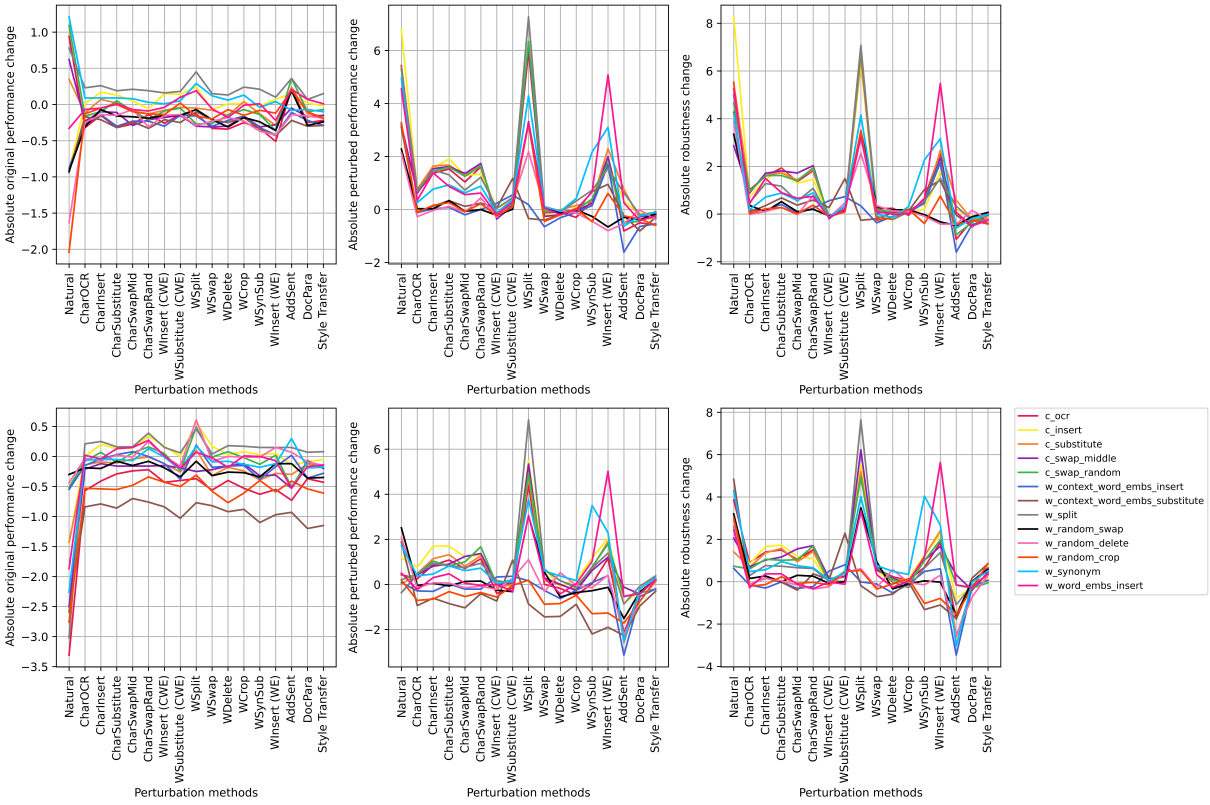

Figure 8: Absolute changes in original and perturbed performance (F1), as well as the robustness of `deberta-large` under natural and various synthetic noises, following retraining with each synthetic perturbation. The upper row and the bottom row illustrate the results on the SQuAD 1.1 and SQuAD 2.0 format test sets, respectively.

Attack	Answered Correctly
CharOCR	64
CharInsert	70
CharSubstitute	56
CharSwapMid	60
CharSwapRand	52
WInsert (CWE)	64
WSubstitute (CWE)	48
WSplit	74
WSwap	60
WDelete	60
WCrop	68
WSynSub	62
WInsert (WE)	58
AddSent	28
DocPara	48
Style Transfer	46

Table 11: The percentage (%) of adversarial QA instances correctly labelled by humans for each synthetic perturbation method.