Consistency of LLMs to Comparative Statements in Mathematical Reasoning Tasks

Aidan San[♢], **Daniel Son**[♢], **Xiaodong Liu**[♣], **Yangfeng Ji**[♢]
[♢]University of Virginia, [♣]Microsoft Research
{aws9xm,gtb4ua,yangfeng}@virginia.edu
xiaodl@microsoft.com

Abstract

Large language models (LLMs) have the potential to significantly expand access to quality education through applications such as mathematics tutoring. However, a key challenge is that student writing often contains redundancies, and prior research has shown that LLMs can be sensitive to such irrelevant information. This raises a critical research question: How consistent are LLMs when faced with extraneous comparative statements? To address this, we propose a systematic framework for evaluating LLM consistency. Our approach involves a hybrid strategy that integrates template-based and model-based methods to generate comparative statements (e.g., "`One of the apples was tastier than average`") and insert them into mathematical reasoning problems. The merit of our approach lies in its systematic and automated nature, enabling rigorous assessment across various models and datasets. Conducting experiments on the GSM8K, AQuA, and Hendrycks MATH benchmarks with a suite of open-source LLMs, we highlight two key results. First, LLM accuracy can drop by over 30% when presented with these statements. Furthermore, we uncover a trade-off between the diversity of the generated statements and the magnitude of the performance drop, where less diverse and more repetitive perturbations lead to greater accuracy degradation.

Keywords: consistency, large language models, evaluation, mathematical reasoning

1. Introduction

The integration of large language models (LLMs) into educational tools is accelerating, with mathematics tutoring emerging as a key area of application (Miller and DiCerbo, 2024; Macina et al., 2025; Gupta et al., 2025). LLM-based tutors offer a promising solution to the high cost and limited availability of traditional tutoring, making academic support more accessible to students.

However, for these tutors to be effective, they must be robust to the natural writing styles of students, which often include redundancies and extraneous details. This is a critical concern, as prior work has shown that the performance of LLMs can degrade when prompts contain irrelevant information (Mirzadeh et al., 2025). Prior work (Mirzadeh et al., 2025) adopts a more manual methodology to examine LLM performance in the presence of various forms of irrelevant information. Our approach focuses on a specific form of extraneous information: comparative statements. To better understand and address this vulnerability, this paper introduces a systematic approach increasing the scale of evaluation from prior work. We formalize a template for these problematic comparative statements—"`One of the {entities} was {comparator} than average`"—to rigorously evaluate their effect on LLM performance. An illustrative example is provided in Figure 1.

This study systematically evaluates the consistency of several open-source LLMs from the Phi, Gemma, Llama, Qwen, and Mistral families when their input is perturbed with comparative statements. To do this, we generate these statements and insert them into mathematical word problems, assessing model consistency by comparing performance on the original and modified problems.

We explore several methods for generating

Prompt: James and Lisa ordered 2 small pizzas. Each pizza has 6 slices. James ate 2/3 of all the slices. One of the slices is chewier than average. How many slices of pizza did James eat?

Llama-3: James ate 2/3 of all the slices, so he ate 2/3 * 12 slices = «2/3*12=8»8 slices. One of the slices is chewier than average, so he did not eat that slice. Therefore, he ate 8 - 1 = «8-1=7»7 slices. #### 7

Figure 1: In the example, we can see an inconsistent model response from the Llama-3 model. When the statement is not inserted, the model responds with the correct answer "8." After the comparative statement "One of the slices is chewier than average." was inserted, the model changes its response and responds with an incorrect answer of "7". The comparative statement should not affect the final answer, but the model changes its response.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 4482–4496
11–16 May 2026.

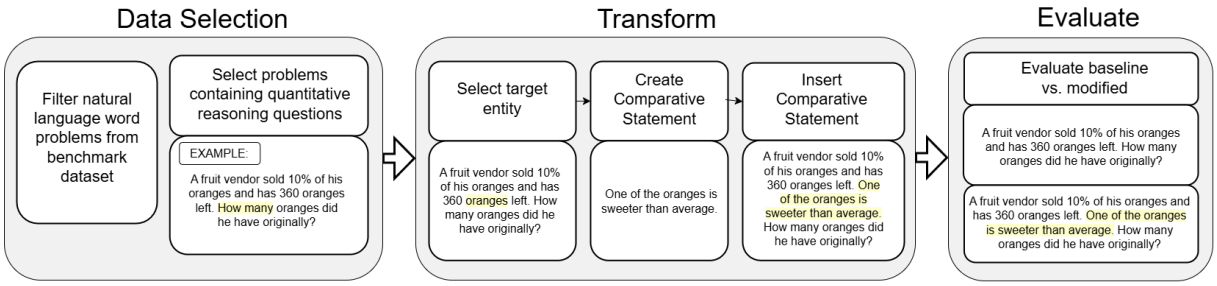

Figure 2: Our process of selecting an example from a mathematical reasoning benchmark, then transforming and evaluating it using our method.

the comparative statements. Our primary approach is template-based, where we populate the structure "`One of the {entities} was {comparator} than average.`" To fill the template, we test three techniques: a fixed baseline using a single common comparator, a masked language model (RoBERTa; Liu et al. 2019), and a prompted autoregressive LLM (Gemma; Team et al. 2024b). In addition, the discussion section investigates a free-form generation approach that does not rely on a predefined template.

Our experiments are conducted on three benchmark mathematics datasets: GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), and Hendrycks MATH (Hendrycks et al., 2021). We find that the insertion of these statements can cause accuracy to drop by over 30%, revealing significant inconsistencies in model behavior. This trade-off is evident when comparing our different statement generation methods. We measure diversity by calculating the entropy and counting the number of unique comparators generated by each method. The Gemma-based approach, for instance, produces a significantly more diverse set of comparative statements than the RoBERTa-based method. Correspondingly, the more diverse Gemma method results in a smaller average drop in model accuracy (e.g., 20.4% on GSM8K) compared to the less diverse RoBERTa and fixed-baseline methods (21.7% and 23.1% on GSM8K, respectively). This pattern suggests that models are more robust to varied and less frequent irrelevant phrases than to repeated and common phrases.

Our main contributions are as follows:

- We introduce a systematic methodology for evaluating LLM consistency, including several strategies (fixed, template-based, and free-form) for generating irrelevant comparative statements to perturb prompts. Our code is open source[1].
- Through extensive experiments on three math reasoning benchmarks (GSM8K, AQuA, and Hendrycks MATH), we demonstrate that leading open-source LLMs exhibit significant performance degradation, with accuracy dropping by over 30% in some cases, when presented with these statements.
- We identify a clear trade-off between the diversity of the generated statements and the magnitude of the accuracy drop, showing that less diverse, repetitive perturbations cause a greater decrease in performance.

[1] https://github.com/aidansan/consistency-llms-comparative-statements

2. Related Work

2.1. Mathematical Reasoning in LLMs

Mathematics has long been a central component of education and remains essential for students' academic development. Consequently, mathematical reasoning has become a standard benchmark for evaluating large language Models (LLMs) (Achiam et al., 2023; Abdin et al., 2024a; Touvron et al., 2023). A large body of work has been dedicated to developing evaluation methods and benchmarks for mathematical reasoning and improving the performance of LLMs on this task.

Math word problems are one of the most popular tasks for evaluating mathematical reasoning (Cobbe et al., 2021; Hendrycks et al., 2021; Ling et al., 2017; Albalak et al., 2025). For instance, GSM8K (Cobbe et al., 2021) consists of 8,500 grade-school-level questions created by crowdworkers. Other datasets focus on math competitions (Hendrycks et al., 2021; Mao et al., 2024; Balunović et al., 2025). In particular, Hendrycks et al. (2021) created a dataset from the AMC and AIME competitions, providing step-by-step solutions for each problem. More recently, datasets have been developed to test the mathematical reasoning of multimodal language models by combining images with text (Wang et al., 2024; Shi et al., 2024; Lu et al., 2024). Our work focuses on math word problems, using the GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), and Hendrycks MATH (Hendrycks et al., 2021) datasets.

To address the challenges of mathematical reasoning, specialized models and techniques have been developed. For example, MathPrompter (Imani et al., 2023) employs multiple chain-of-thought rationales to solve a single problem in several ways, thereby improving answer confidence. Chen and Lin (2025) utilizes paraphrasing to generate more diverse mathematical training data. ToRA (Gou et al., 2024) leverages tools such as computational libraries and symbolic solvers to enhance performance. While our current work uses base language models not specifically designed for mathematical reasoning, we plan to evaluate these specialized techniques in future work.

2.2. Consistency

Consistency in LLMs can be broadly categorized into two types: single-generation and multiple-generation consistency. Single-generation consistency evaluates whether a single output is internally coherent or consistent with a given reference. For instance, factual consistency requires that the generated text is truthful relative to a knowledge base (Fabbri et al., 2021; Wang et al., 2020; Tam et al., 2023; Luo et al., 2025). Other forms of single-generation consistency include conceptual consistency, which examines the alignment between a model's outputs and its internal knowledge (Sahu et al., 2022), and behavioral consistency, which assesses the same alignment from a behavioral perspective (Zheng et al., 2024).

Multiple-generation consistency, on the other hand, assesses whether a model maintains consistent behavior across multiple outputs generated from varied inputs. Examples include positional consistency, where a model's response should not change when the order of model's inputs are shuffled (Li et al., 2023; Tang et al., 2023; Zheng et al., 2023). Another example is multilingual consistency, which evaluates if a model produces the same prediction for the same input provided in different languages (Ohmer et al., 2023). Similarly, paraphrase consistency tests whether a model gives the same output for an original and a paraphrased input (Raj et al., 2023a,b; Nalbandyan et al., 2025). Our work introduces a new dimension to this area by focusing on consistency under the insertion of comparative statements, specifically assessing whether a model yields consistent predictions between an original text and a version augmented with an irrelevant comparative statement.

Beyond evaluation, consistency is also a technique to improve the decoding process (Wang et al., 2023b; Chen et al., 2023; Aggarwal et al., 2023). For example, self-consistency (Wang et al., 2023b) samples multiple reasoning paths from a model and marginalizes over them to determine the final answer. Furthermore, consistency can be leveraged for downstream tasks such as fact-checking (Manakul et al., 2023; Lucas et al., 2023; Ma et al., 2025).

2.3. Adversarial Robustness

Adversarial robustness research examines the vulnerability of LLMs to various adversarial manipulations. One line of this research focuses on perturbations at the character, word, and sentence levels. For example, Wang et al. (2023a); Fan and Tao (2024) evaluate LLM robustness using the Adversarial GLUE dataset (Wang et al., 2022), while Zhu et al. (2024) generates custom perturbations with various libraries instead of relying on a pre-existing benchmark. Other approaches (Xiao et al., 2025; Agrawal et al., 2025) utilize search methods to find effective adversarial perturbations.

Another area of adversarial robustness is prompt injection, where malicious instructions are embedded in the input to make the model perform unintended actions (Perez and Ribeiro, 2022; Liu et al., 2024a,b; Li et al., 2024; Cantini et al., 2025). While our method also involves inserting text, our goal is fundamentally different. Adversarial attacks aim to test model behavior with edge cases, whereas our study evaluates model consistency in response to irrelevant but non-malicious statements, which is more common in collaborative settings, such as using LLMs as tutors. Generally, when a student is interfacing with an LLM tutor, they may unintentionally include irrelevant information when asking questions to the model, but the student is not attempting to stress-test or attack the model.

3. Methods

3.1. Overview

Our method begins with an existing benchmark dataset of mathematical word problems. For each problem, we construct a comparative statement and insert it into the question field to create a modified version of the question. The language model is then evaluated on both the original problem and its modified counterpart containing the comparative statement, and the two responses are compared.

We outline the main steps of our method below:

1. Collect quantitative reasoning word problems from benchmark datasets

2. Create comparative statements using model and template-based approach and insert them into original examples

3. Pass the original and modified examples through an evaluation model and compare results

3.2. Original Question Collection

Natural Language Word Problems. To evaluate LLM consistency, we first identify questions suitable for the insertion of comparative statements. While most examples in the GSM8K dataset are concrete natural language word problems, making them appropriate candidates, the Hendrycks MATH and AQuA datasets contain a significant number of problems that are not applicable. Many problems from these datasets are too abstract. To address this, we use an LLM-based filtering procedure to distinguish concrete natural language word problems from more formulaic or abstract examples. The prompt for this filtering is included in the appendix. This step allows us to exclude questions that are not well-suited for our analysis. Table 1 provides examples of both types of problems.

Quantitative Reasoning Questions. For studying comparative statements, our study specifically targets *quantitative reasoning* word problems, which present a scenario and ask for a numerical quantity, often with the phrase "how many." We use a string-matching heuristic to select all questions containing this phrase. The final size of our filtered dataset is detailed in Table 2.

3.3. Comparative Statement Creation

We employ a template-based approach to create comparative statements for insertion. These statements consist of two main components: an "entity" and a "comparator." The entity refers to the noun being compared, while the comparator is the phrase used to draw the comparison. As illustrated in Figure 3, the statements follow the template: "`One of the {entity} was {comparator} than average`." This statement is inserted directly before the question sentence.

> Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. One of the **clips** was **larger** than average. How many clips did Natalia sell altogether in April and May?

Figure 3: An example of a comparative statement. In this example, the entity is "clips" and the comparator is "larger."

To fill in the entity, we use spaCy (Honnibal et al., 2020) dependency parsing to find the main entity of the question. To fill in the comparator, we consider three approaches: a fixed baseline, a masked language model and a prompt-based autoregressive model. Here we describe each approach in detail:

Fixed. For our baseline method, we always insert the most common comparator word: "larger."

Masked Language Model (MLM). For our MLM method, we use a masked language model to select the highest-probability candidate as the comparator word. We refer to this method as **RoBERTa.**

Large Language Model (LLM). For our LLM method, we prompt an instruction-tuned LLM to fill in the template. The prompt is included in the appendix. We refer to this method as **Gemma.**

4. Experiment Settings

4.1. Datasets

We conduct our experiments on three mathematical reasoning datasets composed of word problems, due to their popularity and mix of question sources. When evaluating each model, we utilize the training splits of each dataset due to their larger size (we do not need to reserve the training split for training since we do not actually finetune any models).

GSM8K Following Mirzadeh et al. (2025), we use the GSM8K dataset (Cobbe et al., 2021), which contains 8.5K grade-school-level math problems focused on topics up to pre-algebra. The dataset was constructed by crowdworkers who used GPT-3-generated seed questions to create both the problems and their natural language solutions. To ensure correctness, all problems were independently re-solved by different crowdworkers.

AQuA The AQuA dataset (Ling et al., 2017) contains 100K multiple-choice math questions. Crowdworkers created the dataset by drawing inspiration from 34K seed questions to write new, similar problems. In this process, each worker reviewed five seed questions, selected one, and generated a related question.

Hendrycks MATH (HMATH) The Hendrycks MATH dataset (Hendrycks et al., 2021) features 12.5K problems from high school mathematics competitions, including the AMC 10, AMC 12, and AIME. Each problem is formatted in LaTeX and is annotated with a difficulty rating from 1 to 5.

4.2. Models

Insertion Creation Models. For our MLM-based fill-in method, we use RoBERTa (Liu et al., 2019) due to its widespread use. For our LLM-based approach, we evaluated seven different models and selected the Gemma2-9B instruction-tuned model (Team et al., 2024b).

Question	Score
Tim has 350 pounds of cement in 100, 50, and 25 pound bags. He has an equal number of each size bag. How many bags of cement does Tim have?	4.9099
$\|x+3\| - \|4-x\| = \|8+x\|$ How many S solutions will this equation have?	-2.2476

Table 1: Examples of concrete natural language word problems versus abstract questions from the AQuA dataset. The first question is an example of a concrete question, and the second is abstract. Model scores below zero are filtered-out.

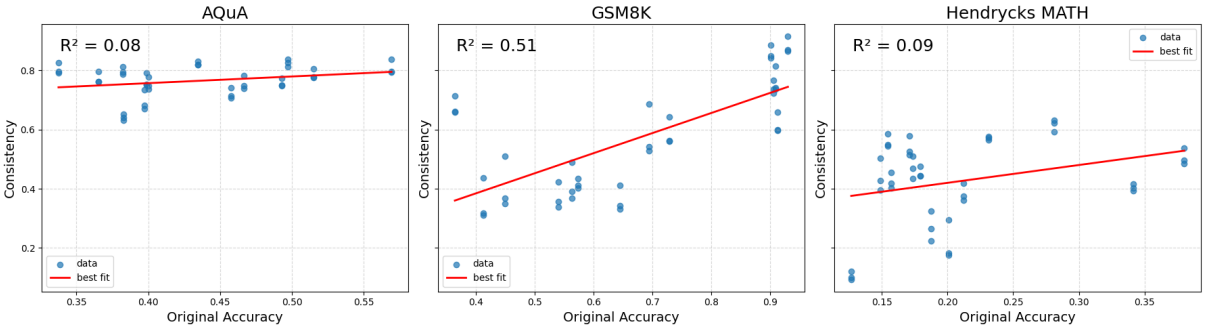

Figure 4: Relationship between original accuracy and consistency across datasets.

Dataset	Original Size	Filtered Size
AQuA	97.5K	3,618
GSM8K	7.5K	1,788
HMATH	12.5K	363

Table 2: The sizes of each benchmark dataset before and after filtering

Evaluation Models. To perform evaluations we use the EleutherAI evaluation framework (Biderman et al., 2024) with 8 fewshot examples. We evaluate the following models: Phi-2, Phi-3 (Abdin et al., 2024a), Phi-4 (Abdin et al., 2024b), Gemma (Team et al., 2024a), Gemma 2 (Team et al., 2024b), Gemma 3 (Team et al., 2025), Llama3 (Grattafiori et al., 2024), Llama3.1, Qwen2.5 (Qwen et al., 2025), and Mistral-v0.3 (Jiang et al., 2023).

4.3. Metrics

To compute **consistency**, we adopt the modified consistency metric introduced by Raj et al. (2023a), which builds upon the formulation of Elazar et al. (2021). Specifically, for each example, if the model produces the same answer y_i for both the baseline and augmented versions of the question, it receives a score of 1; otherwise, it receives a score of 0.

$$f(y_i, y_j) = \begin{cases} 1, & \text{if } y_i = y_j \\ 0, & \text{if } y_i \neq y_j \end{cases}$$

In addition to consistency, we also measure the model's original and modified **accuracy** (before and after insertion of the comparative phrase) and their difference, the accuracy drop, to quantify performance degradation.

4.4. Data Filtering

To ensure the quality of our experimental data, we applied a few final filtering steps. First, since comparative statements are often incompatible with units of measurement (e.g., "`One of the hours was larger than average`"), we removed questions where the main entity was a unit such as "hours," "feet," or "pounds." Second, in the few instances where a generation method failed—for example, when a sentence exceeded RoBERTa's context length—we excluded the corresponding problem across all methods to ensure a fair comparison. Finally, we filtered out cases where spaCy incorrectly extracted the word "more" as the entity in questions like "`How many more cards ...`", addressing a recurring parsing error.

5. Results

5.1. Best Performing Models

Highest Accuracy and Consistency. In the development of LLM-based tutors, identifying models with high consistency and accuracy is crucial for ensuring reliable end-user results. We find that Phi-4 is overall the best-performing model when measuring consistency across nine settings (three insertion methods for three datasets) and measuring accuracy across the three datasets.

Table 4 shows that Phi-4 and Gemma-3-12B are the most consistent models, performing well in 4 out

Insertion Creation Method	Fixed			Gemma			RoBERTa		
Metric	acc_{orig}	acc_{mod}	$cons.$	acc_{orig}	acc_{mod}	$cons.$	acc_{orig}	acc_{mod}	$cons.$
Eval. model									
Llama-3-8B	0.540	0.375	0.356	0.540	0.366	0.422	0.540	0.364	0.338
Llama-3.1-8B	0.563	0.401	0.391	0.563	0.419	0.489	0.563	0.386	0.369
Llama-3.1-8B-Instruct	0.912	0.609	0.599	0.912	0.657	0.658	0.912	0.608	0.596
Mistral-7B-v0.3	0.413	0.319	0.310	0.413	0.315	0.437	0.413	0.323	0.318
Qwen2.5-7B	0.905	0.753	0.737	0.905	0.773	0.767	0.905	0.737	0.724
Qwen2.5-7B-Instruct	0.365	0.328	0.660	0.365	0.326	0.713	0.365	0.327	0.662
Gemma-7b	0.573	0.448	0.413	0.573	0.408	0.433	0.573	0.418	0.403
Gemma-2-9B	0.729	0.571	0.560	0.729	0.570	0.642	0.729	0.573	0.562
Gemma-3-12B-pt	0.695	0.592	0.542	0.695	0.603	0.687	0.695	0.571	0.529
Gemma-3-12B-it	0.900	0.828	0.842	0.900	0.853	0.886	0.900	0.833	0.848
Gemma-3-4B-pt	0.450	0.361	0.367	0.450	0.364	0.510	0.450	0.341	0.351
Phi-2	0.645	0.329	0.344	0.645	0.358	0.412	0.645	0.315	0.331
Phi-3-medium-4k-Instruct	0.909	0.729	0.737	0.909	0.790	0.815	0.909	0.733	0.742
Phi-4	0.930	0.864	0.870	0.930	0.884	0.915	0.930	0.862	0.866

Table 3: Results on GSM8K comparing three insertion creation methods across all evaluation (eval.) models. acc_{orig} refers to the accuracy before inserting the comparative phrase, and acc_{mod} refers to the accuracy after inserting the comparative phrase. $cons$ refers to the consistency between the original and modified accuracy. Original accuracy is the same across all insertion methods, because no comparative statement has been inserted. Phi-4 is the best-performing model. (Gemma is used as both an insertion method as well as an evaluation model.)

Dataset	Insert.	Acc. Best	Cons. Best
GSM8K	Fixed	Phi-4	Phi-4
	Gemma	Phi-4	Phi-4
	RoBERTa	Phi-4	Phi-4
AQuA	Fixed	Phi-4	Gemma-3-12B
	Gemma	Phi-4	Phi-4
	RoBERTa	Phi-4	Gemma-2-9B
HMATH	Fixed	Qwen2.5-7B	Gemma-3-12B
	Gemma	Qwen2.5-7B	Gemma-3-12B
	RoBERTa	Qwen2.5-7B	Gemma-3-12B

Table 4: Highest original accuracy (acc.) and consistency (cons.) models for each dataset and insertion creation method (insert.). Phi-4 and Qwen2.5 are the highest accuracy models. Phi-4 and Gemma-3 are the most consistent models.

of 9 settings. For accuracy, Phi-4 achieves the highest accuracy on the GSM8K and AQuA datasets, while Qwen2.5-7B excels on the Hendrycks MATH dataset. Therefore, Phi-4 represents a generally safe choice for a consistent and accurate math reasoning model. However, since performance can be dataset-dependent, it is important to evaluate various models before deployment. Detailed results for the GSM8K dataset are available in Table 3.

Accuracy vs. Consistency. A natural question is whether more accurate models are also more trustworthy. We investigate this by asking: *Are more accurate models also more consistent?* As shown in Figure 4, there is a positive correlation between accuracy and consistency across all datasets. Among the three, the GSM8K dataset exhibits the strongest correlation ($R^2 = 0.51$), while the Hendrycks MATH dataset shows the weakest ($R^2 = 0.08$). Although the correlation is weak for the AQuA and Hendrycks MATH datasets, our findings suggest that improving model accuracy generally leads to higher consistency. However, there may also be value in specifically training for consistency, particularly in contexts where it is critical.

Benefits of Instruction Tuning. Instruction tuning is a common post-training step for most large language models. One key question is if the instruction tuning process also improves consistency in language models. We find that this effect is dependent on dataset. We focus on three models where we perform evaluations with both the pretrained models as well as their instruction-tuned counterparts: Qwen2.5-7B, Gemma-3-12B, and Llama3.1-8B. On the GSM8K dataset, consistency significantly increases from the pretrained to instruction tuning models for the Gemma and Llama models. On the other hand, consistency decreases across the board on the Hendrycks MATH and AQuA datasets. One possible explanation is that GSM8K data was used for instruction tuning of these models, but AQuA and Hendrycks MATH data was not. For Qwen2.5-7B, there is a different trend, in almost all cases, instruction tuning decreases the consistency of Qwen.

5.2. Comparing Insertion Approaches

Insertion Method Diversity. To thoroughly evaluate the consistency of LLMs with comparative statements, it is essential to test a diverse range of such statements, as models may be highly consistent with one type of statement but sensitive to others. As shown in Figure 7, the generated replacements are often dominated by the most frequent substitution. For both Gemma and RoBERTa, the most common replacement occurs more than twice as often as the second most common. Between the two models, Gemma exhibits greater diversity in its replacements. Specifically, as illustrated in Figure 7, RoBERTa uses "larger" in over 50% of cases, whereas Gemma distributes its replacements more evenly. Furthermore, as shown in Table 5, Gemma has a higher entropy and generates more unique phrases across all datasets, confirming its greater diversity. We explore methods for increasing the diversity of generated phrases in Section 6.

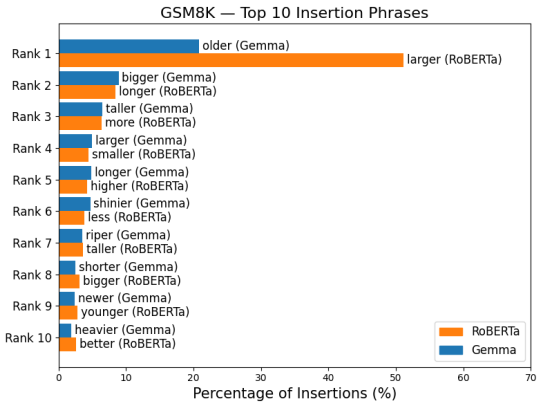

Figure 5: Most common inserted phrases for the RoBERTa vs. Gemma insertion method. The Gemma model produces a much more uniform distribution of insertions compared to the RoBERTa model where more than half of all insertions are the word "larger." There are similar results for the AQuA and Hendrycks MATH datasets.

Dataset	Insert.	Unique	Entropy
AQuA	Gemma	512	6.379
AQuA	RoBERTa	32	3.518
GSM8K	Gemma	253	5.479
GSM8K	RoBERTa	34	2.870
HMATH	Gemma	128	5.982
HMATH	RoBERTa	29	3.272

Table 5: Phrase diversity and entropy across datasets and insertion models. Gemma consistently has higher entropy and more unique phrases implying more diverse insertion phrases.

Dataset	Insert.	Cons.	Acc. Drop
AQuA	Fixed	0.756	7.5%
	Gemma	0.786	7.5%
	RoBERTa	0.753	8.0%
GSM8K	Fixed	0.552	21.7%
	Gemma	0.628	20.4%
	RoBERTa	0.546	23.1%
HMATH	Fixed	0.410	21.9%
	Gemma	0.459	20.7%
	RoBERTa	0.409	20.9%

Table 6: Average consistency (cons.) and relative accuracy (acc.) drops. RoBERTa and the fixed method have the highest drops in accuracy.

Insertion Method Consistency. As shown in Table 6, the lowest consistency and highest relative accuracy drop are observed with the fixed method on the AQuA dataset, and with the RoBERTa method on the GSM8K and Hendrycks MATH datasets. The accuracy drop on the GSM8K dataset is visualized in Figure 6.

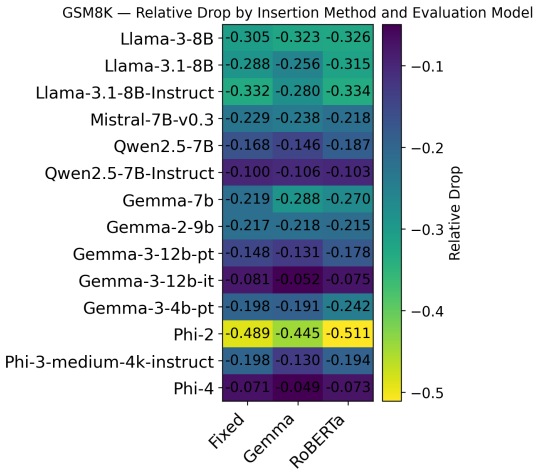

Figure 6: A heatmap of accuracy drop on the GSM8K dataset. RoBERTa overall has the highest accuracy drop.

Quality Annotation. To ensure the reliability of our method, the authors annotated 100 examples from each dataset, for a total of 300 examples. We annotated for two main aspects: change in meaning and fluency. Change in meaning is defined as *an insertion which may change the answer to the problem* and fluency is defined as *a grammatical and logically coherent insertion*. As can be seen from Table 8, meaning is well-preserved across all fill-in methods in the majority (greater than 90%) of examples.

Insert.	Error Type	Problem
Gemma	fluency	There are 40 ducks in a pond. 50 percent of the ducks are Muscovy and 30 percent of the Muscovies are female. One of the ducks is Muscovy ducks than average. How many female Muscovy ducks there?
RoBERTa	meaning	Cora started reading a 158-page book on Monday, and she decided she wanted to finish it by the end of Friday. She read 23 pages on Monday, 38 pages on Tuesday, and 61 pages on Wednesday. She knows she will have time to read twice as much on Friday as she does on Thursday. One of the pages is longer than average. How many pages does she have to read on Thursday to finish the book in time?

Table 7: Example errors from our method on the GSM8K dataset. In the Gemma example, there is an issue with fluency, because "Muscovy ducks" is not a grammatical comparator word. In the RoBERTa example, the meaning of the question slightly changes, because if one of the pages is longer, it may take more time to read that page.

Fill-In	Meaning Preserved	Fluent
Fixed	91.67%	84.67%
Gemma	92.67%	81.00%
RoBERTa	90.67%	80.66%

Table 8: Comparison of annotated meaning preservation and fluency across the three insertion creation methods.

We present examples of quality errors identified in our method in Table 7. In the example from Gemma, we observe a fluency issue: the generated sentence "`One of the ducks is Muscovy ducks than average.`" is ungrammatical because *Muscovy ducks* is not a valid comparative term. In contrast, the example from RoBERTa demonstrates a minor meaning-preservation error. The question concerns reading speeds, but stating that one of the pages is "longer than average" may alter the answer to the question, as a longer page could take more time to read.

6. Discussion

Statement Insertion Position. Past work (Li et al., 2023; Tang et al., 2023; Zheng et al., 2023) has found that LLM consistency is sensitive to the order and positioning of text. During further exploratory studies, we found that the location of the inserted phrase is extremely important to model consistency. When inserted before the "How many" phrase (e.g., `One of the fishes is larger than average. How many fishes did Johnny catch?`), consistency is relatively lower compared to when inserted at the end of the example (e.g., `How many fishes did Johnny catch? One of the fishes is larger than average`). With the Llama-3.1 model, when the phrase is inserted after the "How many" phrase, consistency is 59.2% compared to 48.9% when inserted before the "How many" phrase. This represents a drop of more than 10% in consistency depending on the insertion position. Our findings align with past work stating that LLM consistency is sensitive to positioning of text.

Free-form Generation. In addition to template-based methods, we also wanted to explore allowing the LLM to freely generate the entire comparative statement. One example of such a generated statement is "`The bag of apples weighed less than the bag of oranges.`" We observe a similar trade-off between diversity and accuracy drop, as seen previously. We find that using this free-form generation method results in 25% more unique phrases over the template-based prompting method, and a smaller degradation in Llama-3.1 accuracy—7.65% compared to 25.6% originally.

Entity Selection. We also explored the consistency of comparative statements when selecting other entities. In place of choosing the main entity targeted by the "How many" phrase, we instead selected a random proper noun from the example. For instance, in the example "`How many marbles does Emily have now?`", we created the phrase "`Emily is happier than average.`" With the Llama-3.1 model, we find that there is a smaller relative accuracy drop of only 8.18% compared to 25.6% when using the original entity.

Improving Diversity of Comparative Statements. To increase the diversity of comparative statements, we also wanted to explore alternate ways of selecting the comparator word. Instead of picking the first option proposed by the LLM, we randomly sampled from the first three options generated by the LLM. When selecting from the first three options, there were 49.8% more unique phrases, and entropy increased from 5.479 to 6.467. Model consistency remained nearly unchanged, decreasing slightly

from 48.9% to 48.1%. Based on these results, one possible future direction would be to explore other sampling methods to improve the diversity of our method.

7. Conclusion

This paper introduces a systematic framework for evaluating the consistency of large language models (LLMs) on comparative statements. Our evaluation across a diverse set of open-source models reveals that, under certain conditions, LLMs can suffer a performance degradation of over 30%. We identify a clear trade-off between the diversity of comparative statements and accuracy degradation, where less diverse and more repetitive perturbations—such as those generated by the RoBERTa-based method—cause a greater drop in performance. Future work will investigate whether these findings extend to tasks beyond mathematical reasoning and to closed-source models.

8. Bibliographical References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, et al. 2024a. Phi-3 technical report: A highly capable language model locally on your phone.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024b. Phi-4 technical report.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms.

Aryan Agrawal, Lisa Alazraki, Shahin Honarvar, and Marek Rei. 2025. Enhancing llm robustness to perturbed instructions: An empirical study.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges.

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. 2025. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2025. Matharena: Evaluating llms on uncontaminated math competitions.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. 2024. Lessons from the trenches on reproducible evaluation of language models.

Riccardo Cantini, Alessio Orsino, Massimo Ruggiero, and Domenico Talia. 2025. Benchmarking adversarial robustness to bias elicitation in large language models: scalable automated assessment with llm-as-a-judge. *Machine Learning*, 114(11).

Shuguang Chen and Guang Lin. 2025. Llm reasoning engine: Specialized training for enhanced mathematical reasoning.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation.

Xiaojing Fan and Chunliang Tao. 2024. Towards resilient and efficient llms: A comparative study of efficiency, performance, and adversarial robustness.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. Tora: A tool-integrated reasoning agent for mathematical problem solving.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, et al. 2024. The llama 3 herd of models.

Adit Gupta, Jennifer Reddig, Tommaso Calo, Daniel Weitekamp, and Christopher J. MacLellan. 2025. Beyond final answers: Evaluating large language models for math tutoring.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models.

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.

Zekun Li, Baolin Peng, Pengcheng He, and Xifeng Yan. 2024. Evaluating the instruction-following robustness of large language models to prompt injection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 557–568, Miami, Florida, USA. Association for Computational Linguistics.

Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2023. Split and merge: Aligning position biases in large language model based evaluators.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Xiaogeng Liu, Zhiyuan Yu, Yizhe Zhang, Ning Zhang, and Chaowei Xiao. 2024a. Automatic and universal prompt injection attacks against large language models.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2024b. Prompt injection attack against llm-integrated applications.

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. *arXiv preprint arXiv:1907.11692*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts.

Jason Lucas, Adaku Uchendu, Michiharu Yamashita, Jooyoung Lee, Shaurya Rohatgi, and Dongwon Lee. 2023. Fighting fire with fire: The dual role of llms in crafting and detecting elusive disinformation.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2025. Factual consistency evaluation of summarization in the era of large language models.

Jiatong Ma, Linmei Hu, Rang Li, and Wenbo Fu. 2025. Local: Logical and causal fact-checking with llm-based multi-agents. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 1614–1625, New York, NY, USA. Association for Computing Machinery.

Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2025. Mathtutorbench: A benchmark for measuring open-ended pedagogical capabilities of llm tutors.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

Yujun Mao, Yoon Kim, and Yilun Zhou. 2024. Champ: A competition-level dataset for fine-grained analyses of llms' mathematical reasoning capabilities.

Pepper Miller and Kristen DiCerbo. 2024. Llm based math tutoring: Challenges and dataset.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models.

Grigor Nalbandyan, Rima Shahbazyan, and Evelina Bakhturina. 2025. SCORE: Systematic COnsistency and robustness evaluation for large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 470–484, Albuquerque, New Mexico. Association for Computational Linguistics.

Xenia Ohmer, Elia Bruni, and Dieuwke Hupkes. 2023. Separating form and meaning: Using self-consistency to quantify task understanding across multiple senses.

Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models.

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, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Harsh Raj, Vipul Gupta, Domenic Rosati, and Subhabrata Majumdar. 2023a. Semantic consistency for assuring reliability of large language models.

Harsh Raj, Domenic Rosati, and Subhabrata Majumdar. 2023b. Measuring reliability of large language models through semantic consistency.

Pritish Sahu, Michael Cogswell, Yunye Gong, and Ajay Divakaran. 2022. Unpacking large language models with conceptual consistency.

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. 2024. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2023. Evaluating the factual consistency of large language models through news summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5220–5255, Toronto, Canada. Association for Computational Linguistics.

Raphael Tang, Xinyu Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2023. Found in the middle: Permutation self-consistency improves listwise ranking in large language models.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024a. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024b. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2022. Adversarial glue: A multi-task benchmark for robustness evaluation of language models.

Jindong Wang, HU Xixu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, et al. 2023a. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. In *ICLR 2023*

Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models.

Mingxuan Xiao, Yan Xiao, Shunhui Ji, Hanbo Cai, Lei Xue, and Pengcheng Zhang. 2025. Assessing the robustness of llm-based nlp software via automated testing.

Danna Zheng, Danyang Liu, Mirella Lapata, and Jeff Z. Pan. 2024. Trustscore: Reference-free evaluation of llm response trustworthiness.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, and Xing Xie. 2024. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, LAMPS '24, page 57–68, New York, NY, USA. Association for Computing Machinery.

9. Language Resource References

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd and. 2020. spacy: Industrial-strength natural language processing in python.

A. Prompts

A.1. Prompt for natural language word problem filtering

```
<|begin_of_text|><|start_header_id|>
system<|end_header_id|>

Cutting Knowledge Date: December
 ↪  2023
Today Date: 26 Jul 2024

You are a helpful
 ↪  assistant<|eot_id|>
<|start_header_id|>
user<|end_header_id|>

Is the following question a word
 ↪  problem?
Mr. Kramer, the losing candidate in
 ↪  a two-candidate election,
 ↪  received 942,568 votes, which
 ↪  was exactly 47 percent of all
 ↪  votes cast. Approximately what
 ↪  percent of the remaining votes
 ↪  would he need to have received
 ↪  in order to have won at least 50
 ↪  percent of all the votes
 ↪  cast?<|eot_id|>
<|start_header_id|>
assistant<|end_header_id|>

Yes<|eot_id|><|start_header_id|>
user<|end_header_id|>

Is the following question a word
 ↪  problem?
The perimeter of a triangle is 28 cm
 ↪  and the inradius of the triangle
 ↪  is 2.5 cm. What is the area of
 ↪  the triangle?<|eot_id|>
<|start_header_id|>
assistant<|end_header_id|>

No<|eot_id|><|start_header_id|>
user<|end_header_id|>

Is the following question a word
 ↪  problem?
23 people attend a party. Each
 ↪  person shakes hands with at most
 ↪  22 other people. What is the
 ↪  maximum possible number of
 ↪  handshakes, assuming that any
 ↪  two people can shake hands at
 ↪  most once?<|eot_id|>
<|start_header_id|>
assistant<|end_header_id|>

Yes<|eot_id|><|start_header_id|>
user<|end_header_id|>

Is the following question a word
 ↪  problem?
```

```
A football player scores 2 goals in
 ↪  his fifth match thus increasing
 ↪  his average goals score by 0.1.
 ↪  The total number of goals in his
 ↪  5 matches would
 ↪  be<|eot_id|><|start_header_id|>
assistant<|end_header_id|>

Yes<|eot_id|><|start_header_id|>
user<|end_header_id|>

Is the following question a word
 ↪  problem?
How many 9 step paths are there from
 ↪  $E$ to $G$ which pass through
 ↪  $F$?[asy]size(4cm,4cm);int
 ↪  w=6;int h=5;int i;pen
 ↪  p=fontsize(9);for (i=0; i<h;
 ↪  ++i){draw((0,i) -- (w-1,i));}for
 ↪  (i=0; i<w; ++i){draw((i,
 ↪  0)--(i,h-1));}label("G", (w-1,0),
 ↪  SE, p);label("E", (0,h-1), NW,
 ↪  p);label("F", (3,3), NE,
 ↪  p);[/asy]<|eot_id|>
<|start_header_id|>
assistant<|end_header_id|>

No<|eot_id|><|start_header_id|>
user<|end_header_id|>

Is the following question a word
 ↪  problem?
A street has 50 houses on each side,
 ↪  for a total of 100 houses. The
 ↪  addresses on the south side of
 ↪  the street form an arithmetic
 ↪  sequence, as do the addresses on
 ↪  the north side of the street. On
 ↪  the south side, the addresses
 ↪  are 1, 5, 9, etc., and on the
 ↪  north side they are 3, 7, 11,
 ↪  etc. A sign painter paints house
 ↪  numbers on a house for $\$1$ per
 ↪  digit. If he paints the
 ↪  appropriate house number once on
 ↪  each of these 100 houses, how
 ↪  much does he earn?<|eot_id|>
<|start_header_id|>
assistant<|end_header_id|>

Yes<|eot_id|><|start_header_id|>
user<|end_header_id|>

Is the following question a word
 ↪  problem?
85% of a number is added to 24, the
 ↪  result is the same number. Find
 ↪  the number?
<|eot_id|><|start_header_id|>
assistant<|end_header_id|>

No<|eot_id|><|start_header_id|>
user<|end_header_id|>

Is the following question a word
 ↪  problem?
The roots of the equation $x^2+kx+5
 ↪  = 0$ differ by $\sqrt{61}$. Find
 ↪  the greatest possible value of
 ↪  $k$.<|eot_id|>
<|start_header_id|>
assistant<|end_header_id|>

No<|eot_id|><|start_header_id|>
user<|end_header_id|>
```

A.2. Prompt for LLM Template insertion method

```
<bos><start_of_turn>user
Identify the words can be used to
 ↪  replace the word [MASK] in an
 ↪  example without changing the
 ↪  answer to the question.
"At a convention, 16 of 36 delegates
 ↪  arrived with pre-printed name
 ↪  badges. Half of the remaining
 ↪  delegates made their own,
 ↪  hand-written name badges. How
 ↪  many delegates were not wearing
 ↪  name badges?"<end_of_turn>
<start_of_turn>model
[MASK] could be replaced with:
* **more talkative**
* **happier**
* **taller**
* **older**
* **skinnier**<end_of_turn>
<start_of_turn>user
Identify the words can be used to
 ↪  replace the word [MASK] in the
 ↪  example without changing the
 ↪  answer to the question.
"Tabitha and her friends were
 ↪  excited to go to the Christmas
 ↪  parade because they knew candy
 ↪  would be thrown to the crowd
 ↪  from the floats.  Tabitha caught
 ↪  22 pieces of candy.  Stan caught
 ↪  13 pieces.  Julie caught half
 ↪  the amount of candy as Tabitha
 ↪  caught and Carlos caught twice
 ↪  as much candy as Stan. One of
 ↪  the pieces is [MASK] than
 ↪  average. How many pieces in
 ↪  total did the friends
 ↪  catch?"<end_of_turn>
```

```
<start_of_turn>model
[MASK] could be replaced with:
* **more uniquely shaped**
* **more tightly wrapped**
* **rarer**
* **sweeter**
* **heavier**<end_of_turn>
<start_of_turn>user
Identify the words can be used to
 ↪  replace the word [MASK] in the
 ↪  example without changing the
 ↪  answer to the question.
"Mark has a garden with flowers. He
 ↪  planted plants of three
 ↪  different colors in it. Ten of
 ↪  them are yellow, and there are
 ↪  80% more of those in purple.
 ↪  There are only 25% as many green
 ↪  flowers as there are yellow and
 ↪  purple flowers. One of the
 ↪  flowers is [MASK] than average.
 ↪  How many flowers does Mark have
 ↪  in his garden?"<end_of_turn>
<start_of_turn>model
[MASK] could be replaced with:
* **larger**
* **smaller**
* **more fragrant**
* **more colorful**
* **duller**<end_of_turn>
```

A.3. Prompt for Free-form Generation insertion method

```
Write one concise comparative
 ↪  sentence describing {entity} in
 ↪  a typical word-problem context.
It must include a comparative phrase
 ↪  with “than” (e.g., “more than,”
 ↪  “less than,” “earlier than,”
 ↪  “larger than”).
Avoid any new numbers or wording
 ↪  that could affect calculations.

For example:
Entity: books
Insertion: "Some books had more worn
 ↪  covers than the rest."

Entity: {entity}
Insertion:
```

B. Additional Results

Insertion Creation Method	Fixed			Gemma			RoBERTa		
Metric	acc_{orig}	acc_{mod}	$cons.$	acc_{orig}	acc_{mod}	$cons.$	acc_{orig}	acc_{mod}	$cons.$
Eval. model									
Llama-3-8B	0.400	0.366	0.749	0.400	0.368	0.778	0.400	0.364	0.738
Llama-3.1-8B	0.399	0.368	0.752	0.399	0.366	0.792	0.399	0.362	0.750
Llama-3.1-8B-Instruct	0.397	0.353	0.671	0.397	0.367	0.734	0.397	0.363	0.681
Mistral-7B-v0.3	0.337	0.313	0.793	0.337	0.315	0.826	0.337	0.310	0.797
Qwen2.5-7B	0.493	0.455	0.750	0.493	0.456	0.774	0.493	0.444	0.747
Qwen2.5-7B-Instruct	0.467	0.456	0.747	0.467	0.462	0.782	0.467	0.449	0.739
Gemma-7b	0.383	0.336	0.640	0.383	0.335	0.652	0.383	0.334	0.632
Gemma-2-9B	0.434	0.397	0.820	0.434	0.391	0.831	0.434	0.394	0.820
Gemma-3-12B-pt	0.497	0.464	0.827	0.497	0.468	0.837	0.497	0.464	0.811
Gemma-3-12B-it	0.457	0.451	0.713	0.457	0.440	0.740	0.457	0.447	0.707
Gemma-3-4B-pt	0.382	0.345	0.795	0.382	0.341	0.813	0.382	0.345	0.787
Phi-2	0.365	0.328	0.763	0.365	0.324	0.796	0.365	0.320	0.762
Phi-3-medium-4k-instruct	0.515	0.488	0.777	0.515	0.488	0.807	0.515	0.493	0.779
Phi-4	0.569	0.536	0.793	0.569	0.534	0.837	0.569	0.537	0.796

Table 9: Results on AQuA comparing three insertion creation methods across all evaluation (eval.) models.

Insertion Creation Method	Fixed			Gemma			RoBERTa		
Metric	acc_{orig}	acc_{mod}	$cons.$	acc_{orig}	acc_{mod}	$cons.$	acc_{orig}	acc_{mod}	$cons.$
Eval. model									
Llama-3-8B	0.179	0.152	0.444	0.179	0.127	0.477	0.179	0.152	0.444
Llama-3.1-8B	0.174	0.157	0.468	0.174	0.176	0.510	0.174	0.157	0.435
Llama-3.1-8B-Instruct	0.187	0.138	0.223	0.187	0.152	0.325	0.187	0.163	0.264
Mistral-7B-v0.3	0.154	0.121	0.545	0.154	0.118	0.587	0.154	0.116	0.548
Qwen2.5-7B	0.380	0.287	0.485	0.380	0.292	0.537	0.380	0.287	0.496
Qwen2.5-7B-Instruct	0.127	0.083	0.094	0.127	0.088	0.121	0.127	0.091	0.099
Gemma-7b	0.157	0.091	0.419	0.157	0.099	0.455	0.157	0.094	0.402
Gemma-2-9B	0.231	0.196	0.573	0.231	0.201	0.576	0.231	0.201	0.565
Gemma-3-12B-pt	0.281	0.237	0.623	0.281	0.240	0.631	0.281	0.251	0.592
Gemma-3-12B-it	0.212	0.182	0.375	0.212	0.201	0.419	0.212	0.198	0.361
Gemma-3-4B-pt	0.171	0.138	0.515	0.171	0.129	0.579	0.171	0.129	0.526
Phi-2	0.149	0.138	0.397	0.149	0.135	0.504	0.149	0.129	0.427
Phi-3-medium-4k-instruct	0.201	0.138	0.182	0.201	0.146	0.295	0.201	0.124	0.176
phi-4	0.342	0.242	0.402	0.342	0.220	0.416	0.342	0.240	0.394

Table 10: Results on Hendrycks MATH comparing three insertion creation methods across all evaluation (eval.) models.

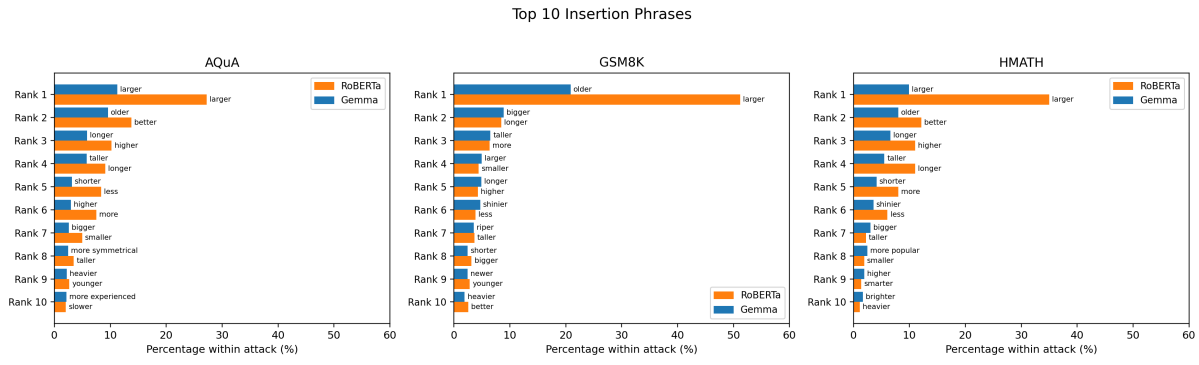

Figure 7: Most common inserted phrases for the RoBERTa vs. Gemma insertion method.