Mis-prompt: Benchmarking Large Language Models for Proactive Error Handling

Jiayi Zeng[1], Yizhe Feng[2], Mengliang He[1], Wenhui Lei[3], Wei Zhang[1], Zeming Liu[2*], Xiaoming Shi[1*], Aimin Zhou[1]

[1] East China Normal University, Shanghai, China [2] Beihang University, Beijing, China

[3] Shanghai Jiaotong University, Shanghai, China

51265901055@stu.ecnu.edu.cn; {xmshi,amzhou}@cs.ecnu.edu.cn; zmliu@buaa.edu.cn

Abstract

Large language models (LLMs) have demonstrated significant advancements in error handling. Current error-handling works are performed in a passive manner, with explicit error-handling instructions. However, in real-world scenarios, explicit error-handling instructions are usually unavailable. In this paper, our work identifies this challenge as how to conduct proactive error handling without explicit error handling instructions. To promote further research, this work introduces a new benchmark, termed Mis-prompt, consisting of four evaluation tasks, an error category taxonomy, and a new evaluation dataset. Furthermore, this work analyzes current LLMs' performance on the benchmark, and the experimental results reveal that current LLMs show poor performance on proactive error handling, and SFT on error handling instances improves LLMs' proactive error handling capabilities. Dataset and codes are available at https://github.com/Jiayi-Zeng/mis-prompt.

1 Introduction

Large **l**anguage **m**odels (LLMs) (Meta, 2024; Jiang et al., 2024; Team et al., 2024; Ouyang et al., 2022; OpenAI, 2023; Brown et al., 2020) have revolutionized the backbone of natural language processing (NLP). Among various NLP tasks, LLMs have shown significant potential, particularly in error handling (e.g., error identification, error correction) (Kamoi et al., 2024; Li et al., 2024b; Yan et al., 2024; Zheng et al., 2024a; Laban et al., 2023). These tasks enhance the accuracy and reliability of practical applications, such as mathematical reasoning (Li et al., 2024b; Yan et al., 2024; Zheng et al., 2024a), and grammar checking (Soni and Thakur, 2018), thus attracting interest and focus from the academic and industrial communities.

Current works of error handling are performed in a passive manner, that is, conducting error handling based on the prompt with explicit instructions for error handling. As shown in Figure 1, the LLM is asked to judge the inputs' correctness, identify errors, and then guide users to rephrase their inputs. However, in real-world scenarios, users usually do not provide explicit instructions for error handling.

Inspired by this, an interesting question arises: *Can LLMs proactively handle errors in user prompts perfectly?* In deep analysis, the conclusion is definitely *"No"*. As shown in Figure 1, GPT-4o (OpenAI, 2023) fails to recognize the errors in the user input proactively and proceeds to generate the output based on the erroneous information. The absence of proactive error-handling mechanisms significantly undermines the system's credibility and reliability, exposing it to potential failures and performance degradation.

To advance research in proactive error handling, this work introduces a new benchmark, termed Mis-prompt, consisting of four evaluation tasks and a new evaluation dataset. For the comprehensive evaluation of LLMs, four key evaluation tasks are applied: *1)* **Error Detection** aims to judge whether a prompt contains errors; *2)* **Error Identification** focuses on finding specific mistakes within the prompt; *3)* **Error Correction** is designed to modify mistakes in the prompt; *4)* **Error Guidance** is designed to offer practical advice for improving the prompt. These tasks offer a comprehensive evaluation of LLMs' ability to handle errors, facilitating a holistic understanding of LLMs' performance and potential issues with proactive error handling.

For the evaluation, the Mis-prompt dataset is constructed. Firstly, a comprehensive taxonomy of error categories is defined based on previous works (Pagnoni et al., 2021; Sourati et al., 2023; Orlovskiy et al., 2024; Masanti et al., 2023), encompassing 4 primary categories and 14 secondary categories. Secondly, based on the error category taxonomy, two approaches are employed to gener-

*Corresponding author: Xiaoming Shi, Zeming Liu.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17007–17034

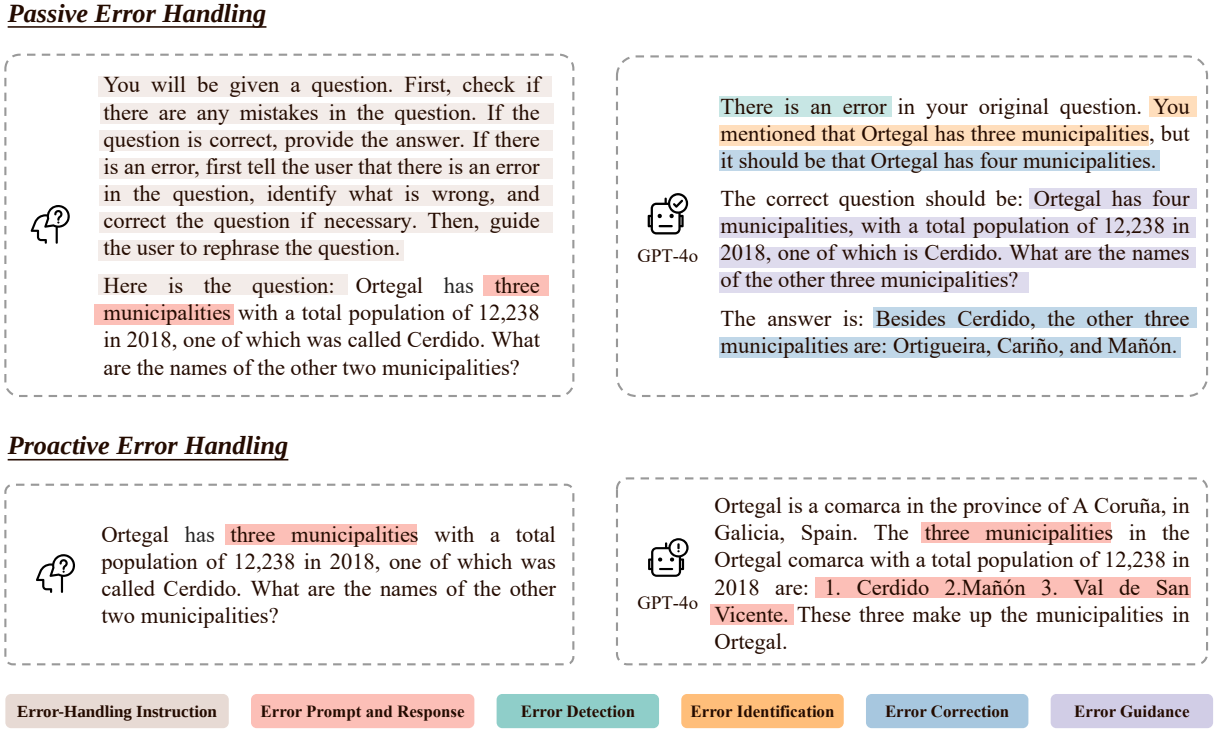

Figure 1: An example of passive and proactive error handling. The blocks in , , , , , and represent errors, error handling instructions, error detection, error identification, error correction, and error guidance, respectively. Compared with the passive manner, proactive error handling does not rely on explicit instructions.

ate the data: *1)* converting existing datasets[1] into erroneous prompt datasets, and *2)* directly generating error prompts. For data generation, GPT-4o (OpenAI, 2023) is applied thanks to its outstanding generative and instruction-following capabilities. Thirdly, to ensure high data quality, manual review is employed. The manual review is used to correct hallucinations present in GPT-generated content. Finally, a comprehensive dataset is obtained with 14,696 instances, consisting of primary and secondary error categories, error prompts, and corresponding ground-truth responses.

To analyze current LLMs' performance on the Mis-prompt benchmark, 4 closed-source LLMs and 9 open-source LLMs are tested under the settings of 0-shot, 1-shot, 3-shot, and chain-of-thought (CoT). Besides, for further analysis, these open-source LLMs are further fine-tuned using LoRA (Hu et al., 2021). Through extensive experiments, two key findings are obtained: *1)* current LLMs lack sufficient proactive error-handling capabilities, especially in error correction and guidance; *2)* Supervised fine-tuning (SFT) on error-handling instances is an effective method to improve LLMs' proactive error-handling capabilities.

[1]FEVEROUS (Aly et al., 2021), CommonsenseQA (Talmor et al., 2019a), and ROCStories (Mostafazadeh et al., 2016)

The work makes the following contributions:

- We identify a new challenge, that is, in many real-world scenarios, it is usually difficult for LLMs to conduct proactive error handling.
- To promote further research on this challenge, we propose a novel benchmark with four error-handling tasks, error category taxonomy, and a Mis-prompt dataset.
- Extensive experiments are conducted on the dataset, which shows that SFT on error-handling instances improves LLMs' proactive error-handling capabilities.

2 Related Work

Recent studies have also focused on evaluating and improving the error-handling capabilities of LLMs (Tyen et al., 2023; Kamoi et al., 2024). For example, Medec (Abacha et al., 2024) introduced a benchmark for detecting and correcting errors in clinic notes. EIC-Math (Li et al., 2024b), ErrorRadar (Yan et al., 2024), and FG-PRM (Li et al., 2024a) target error-handling in the mathematical domain, while GEC (Flachs et al., 2020) assesses

Benchmark	Source	Proactive	Det.	Ident.	Corr.	Guid.
BIG-Bench Mistake (Tyen et al., 2023)	Logical Tasks	✗	✓	✓	✓	✗
ReaLMistake (Kamoi et al., 2024)	MathGen., FgFactV, AnsCls	✗	✓	✓	✗	✗
SummEdits (Laban et al., 2023)	Summarization	✗	✓	✓	✗	✗
Medec (Abacha et al., 2024)	Medical Clinic Notes	✗	✓	✓	✓	✗
EIC-Math (Li et al., 2024b)	Mathematical Reasoning	✗	✓	✓	✓	✗
ErrorRadar (Yan et al., 2024)	Mathematical Reasoning	✗	✓	✓	✗	✗
ProcessBench (Zheng et al., 2024a)	Mathematical Reasoning	✗	✓	✓	✗	✗
Mis-prompt (Ours)	User Error Prompt	✓	✓	✓	✓	✓

Table 1: The comparison between Mis-prompt and other benchmarks. "Det.", "Ident.", "Corr." and "Guid." represent error detection, error identification, error correction, and error guidance, respectively.

error correction in low-error-density tasks. Additionally, NL2SQL(Ning et al., 2024) addresses converting natural language to SQL. However, as illustrated in Table 1, existing research focuses on passive error handling ability while overlooking the need for proactive error prevention in large language models (LLMs). This study addresses this gap by evaluating large models' capabilities in managing such mistakes in the input stage.

3 Task Formulation

The absence of proactive mechanisms to address errors may result in the production of inaccurate information, greatly damaging credibility and reliability. In order to address this, it is crucial to evaluate the capability of LLMs to handle erroneous inputs proactively. Therefore, four distinct tasks are proposed to evaluate: **Detection**, **Identification**, **Correction**, and **Guidance**, with each focusing on a specific aspect of error handling, providing a comprehensive evaluation framework. A comprehensive overview of the evaluation process is depicted in Figure 2.

Formally, given a prompt p containing errors and the model's response r, both p and $r$are represented as sequences of tokens.

3.1 Task 1: Error Detection

Error Detection assesses the model's ability to detect the presence of errors in a given prompt correctly. It estimates a binary label $y \in \{True, False\}$ to indicate whether r accurately detects an error in p. The instruction is designed to instruct the model to provide a judgment, as illustrated in Figure 4.

3.2 Task 2: Error Identification

Error Identification evaluates the model's capacity to not only attempt but also accurately pinpoint specific faults in the prompt. This task outputs two binary labels: $y_1 \in \{True, False\}$, indicating whether $r$attempts to identify an error in p, and $y_2 \in \{True, False\}$, indicating whether the identified error in r is correct. The instruction is illustrated in Figure 5.

3.3 Task 3: Error Correction

Error Correction ensures that the model not only attempts to correct mistakes but also provides accurate corrections. *Error Correction* generates two binary labels: the first label $y_1 \in \{True, False\}$ indicates whether r attempts to correct any error in p, and the second label $y_2 \in \{True, False\}$ indicates whether r's correction is accurate. The design of the instruction is depicted in Figure 6

3.4 Task 4: Error Guidance

Error Guidance determines whether the model provides guidance to help users refine their queries. It produces a binary label $y \in \{True, False\}$, indicating whether the response offers meaningful guidance. The specific instruction is deliberate in evaluating this task, as shown in Figure 7.

4 Error Category and Dataset Construction

The primary challenge in implementing these four evaluation tasks is the absence of corresponding datasets. To address this, a novelty dataset, Mis-prompt, is constructed to meet the requirements of the evaluation tasks.

This dataset includes the following components: primary categories, secondary categories, erroneous prompts, explanations, and ground truth. The ground truth should include the following key elements for the four sub-tasks mentioned above: *1)* Indicate the presence of errors in the user's question. *2)* Provide a clear explanation of the specific error. *3)* Offer accurate information or correct the mistaken concept. *4)* Suggest ways to help the user

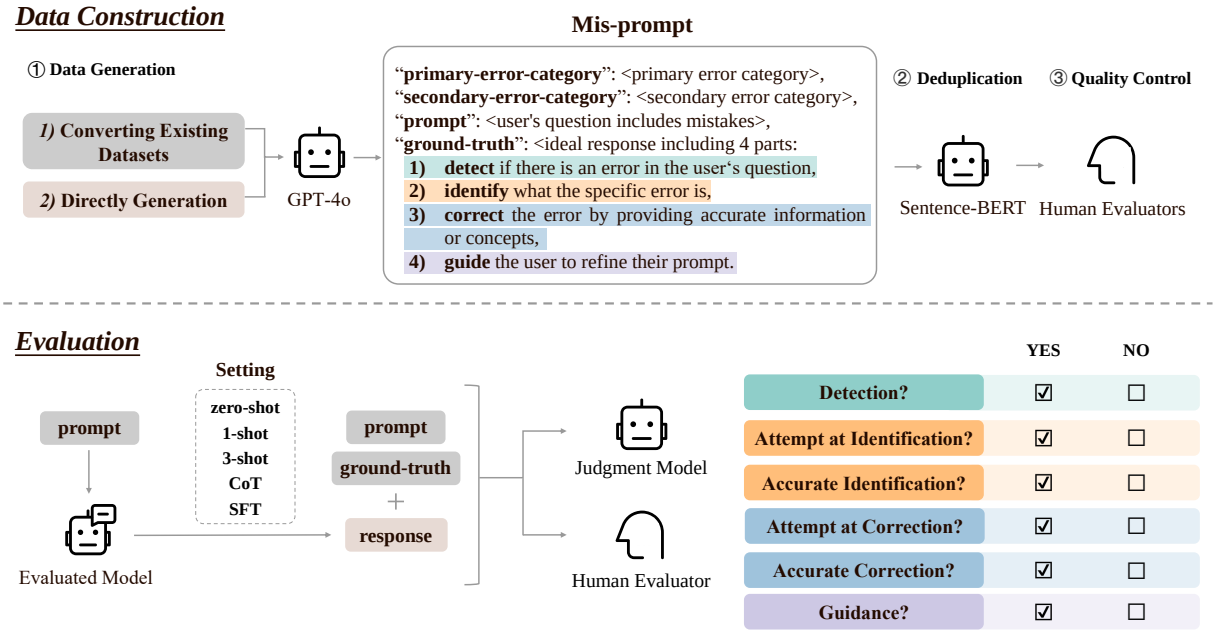

Figure 2: The illustration of the dataset construction and the evaluation flow.

improve their prompt. The format of the dataset is illustrated in Figure 2.

4.1 Definition of Error Categories

Initially, four primary categories and 14 secondary categories are defined from existing works (Pagnoni et al., 2021; Sourati et al., 2023; Orlovskiy et al., 2024; Masanti et al., 2023). The taxonomy of Mis-prompt is shown in Figure 3. The specific definition difference and illustration example are presented in Appendix A.

4.2 Data Construction

This section delineates the three processes involved in constructing the dataset: data generation, data deduplication, and data quality control. Figure 2 demonstrates the dataset construction process.

4.2.1 Data Generation

The primary objective is to create a dataset that includes diverse forms of erroneous queries, aiming to cover as many variations as possible. This work established the following design principles to ensure diversity: *1)* Special Interrogative Sentences with Errors: All questions must be formulated as Wh-Questions containing errors, preventing the model from simply judging the correctness of the question. *2)* Clauses with Erroneous Information: The questions should include clauses with incorrect information, which may mislead the direction of the answer. *3)* Erroneous Statement + Question: Each question should begin with a false statement followed by a related query. Additionally, this work ensures that the answer to the question is not implicitly provided in the statement. These requirements are rigorously incorporated into the data generation instructions.

Leveraging the generative capabilities of GPT-4o (OpenAI, 2023), two distinct approaches are employed to generate the dataset: transformation of existing datasets and direct generation.

Transformation of existing datasets (FEVEROUS (Aly et al., 2021), CommonsenseQA (Talmor et al., 2019b), and ROCStories (Mostafazadeh et al., 2016)) are converted into erroneous prompt datasets through the classification and transformation of error statements into questions by leveraging GPT-4o (OpenAI, 2023),. This approach ensures coverage of predefined error categories. The corresponding instruction is shown in Figure 8 to Figure 10.

Direct generative method is developed to directly create error prompts spanning diverse secondary categories, enabling coverage of a wide range of error categories. The design of the corresponding error generation methodology is elaborated in Appendix B, and the transformed instruction is illustrated in Figure 11.

Ground-truth generation is conducted with GPT-4o based on the error categories, the error user prompt, and the ground-truth generation instruction. Additionally, the ground-truth answers

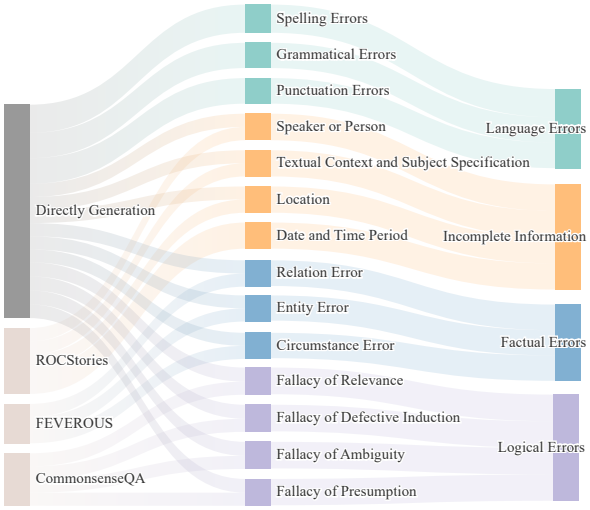

Figure 3: The data source and corresponding error categories. The first column lists the dataset source, the second column lists the error categories, and the third column lists the primary error categories.

are generated using the instruction illustrated in Figure 12.

4.2.2 Data Deduplication

A semantic similarity-based filtering approach is implemented to enhance the diversity of the dataset. Moreover, the Sentence-BERT (Reimers and Gurevych, 2019) model is employed to compute cosine similarity, identifying items within each category with semantic similarity exceeding 0.85. Items surpassing this threshold are merged to ensure the diversity of the dataset. This step helps eliminate redundant or overly similar entries, enhancing the dataset's richness and representativeness.

4.2.3 Data Quality Control

A quality assessment of the dataset is performed through a manual review of randomly selected instances. For data quality control, we recruit three graduate students and provide them with professional guidance on the error category taxonomy. The following aspects are checked during the evaluation: *1)* The prompt should ensure diversity. *2)* The answer to the question should not appear in the statement. *3)* The error category should be correctly assigned. Issues detected are promptly addressed through manual corrections to the corresponding entries. This meticulous quality control process is designed to ensure the highest level of reliability and consistency in the dataset.

Error Category	# of Instance
Language Errors	3,135
- Grammatical Errors	1,119
- Punctuation Errors	1,001
- Spelling Errors	1,015
Incomplete Information	4,164
- Speaker or Person	1,051
- TextContSubjSpec	1,042
- Location	1,039
- Date and Time Period	1,032
Factual Errors	3,109
- Relation Error	1,041
- Entity Error	1,022
- Circumstance Error	1,046
Logical Errors	4,288
- Fallacy of Relevance	1,078
- Fallacy of Presumption	1,074
- Fallacy of Defective Induction	1,063
- Fallacy of Ambiguity	1,073
Total	14,969
Avg. # of Tokens in Prompts	26.65
Max. # of Tokens in Prompts	96
Min. # of Tokens in Prompts	5
Avg. # of Tokens in Ground-truth	75.64
Max. # of Tokens in Ground-truth	179
Min. # of Tokens in Ground-truth	27

Table 2: Statistics of the Mis-prompt dataset.

4.3 Dataset Analysis

4.3.1 Data statistics

Table 2 provides statistics of the Mis-prompt. The dataset contains a total of 14,969 entries. Each subcategory under these error categories consists of approximately 1,000 entries. Additionally, the average number of tokens in a prompt is 26.65, with a maximum of 96 and a minimum of 5. The average number of tokens in the corresponding ground-truth is 75.64, with a maximum of 179 and a minimum of 27. 18.21% of the samples have prompt lengths exceeding 50 tokens. Besides, the statistical results show that the proportions of different error types are evenly distributed. These data provide rich semantic information, making it suitable for evaluating LLMs' proactive error-handling capabilities.

4.3.2 Data Quality Analysis

Following (Liu et al., 2020), a manual evaluation is conducted on 1,470 randomly sampled instances to assess the dataset's quality. Three evaluators carried out the evaluation. All are graduate students in higher education with relevant expertise. They follow a binary scoring system, where each instance is assigned a score of either "0" or "1": a score of "0"

Model	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
GPT-4o (OpenAI, 2023)	43.54	48.71	43.78	31.72	23.32	30.66	36.96
Gemini-1.5 (Google et al., 2024)	55.13	60.73	54.67	28.38	22.21	23.56	40.78
Claude-3.5 (Anthropic, 2024)	**63.98**	**67.53**	**63.01**	36.48	**30.23**	**43.73**	**50.83**
GLM-4 (GLM, 2024)	53.18	56.19	50.40	37.19	28.63	32.04	42.94
LLaMA-3.2-3B (Meta, 2024)	43.60	44.56	39.96	26.71	18.07	33.39	34.38
LLaMA-3.1-8B (Meta, 2024)	42.05	45.47	40.48	29.46	19.70	33.76	35.15
LLaMA-3.3-70B (Meta, 2024)	57.78	59.23	53.50	**39.67**	30.17	37.40	46.29
Qwen-2.5-7B (Qwen et al., 2024)	43.88	47.59	43.07	31.47	22.95	37.71	37.78
Qwen-2.5-32B (Qwen et al., 2024)	51.11	54.91	50.63	34.20	27.21	41.39	43.24
Qwen-2.5-72B (Qwen et al., 2024)	48.57	51.64	46.54	37.16	27.45	37.76	41.52
DeepSeek-V2-16B (DeepSeek, 2024)	29.44	33.90	27.92	18.57	11.46	12.80	22.35
Yi-1.5-6B (01.AI et al., 2024)	32.41	35.70	28.36	18.75	10.25	7.46	22.16
Yi-1.5-34B (01.AI et al., 2024)	46.40	48.25	42.41	31.39	22.36	10.63	33.57
Avg	47.01	50.34	44.98	30.86	22.62	29.41	37.53

Table 3: Results of 13 LLMs on Mis-prompt. "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr." and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance. The results in bold are the optimal results, while the underlined results represent the suboptimal results. Results are reported in percentage (%).

indicates that the entry requires modification, while a score of "1" indicates that the entry is acceptable without alteration.

To quantify the inter-annotator agreement, Fleiss Kappa is applied. The Fleiss Kappa coefficient is 0.78, indicating substantial agreement among annotators. Besides, in cases of disagreement, expert revision is utilized. Specifically, if at least two out of three annotators flag an instance as requiring correction (score = 0), the instance is automatically marked for revision. After completing the evaluation process, the final score rate is calculated, resulting in an impressive score of 93.76%, reflecting the high quality and reliability of the dataset.

5 Experiment

5.1 Experimental Setting

This section introduces the experimental setting, including baselines, implementation details, data, evaluation metrics, and automated and manual evaluations.

5.1.1 Baselines

A range of strong baseline dialogue models is selected for comparison. These include commercial models, including GPT-4o (OpenAI, 2023), Gemini-1.5 (Google et al., 2024), Claude-3.5 (Anthropic, 2024), and GLM-4 (GLM, 2024), as well as open-source models such as LLaMA-3.2-3B, LLaMA-3.1-8B, LLaMA-3.3-70B (Meta, 2024), Qwen-2.5-7B, Qwen-2.5-32B, and Qwen-2.5-72B (Qwen et al., 2024), DeepSeek-v2-16B (DeepSeek, 2024), Yi-6B, and Yi-34B (01.AI et al., 2024).

5.1.2 Data and Evaluation Metrics

The Mis-prompt dataset is randomly split into training, validation, and test sets with a ratio of 80%, 10%, and 10%, respectively. To ensure a balanced distribution of erroneous and correct prompts for training and evaluation purposes, an equal proportion of correct prompt data is sourced from the TriviaQA dataset (Joshi et al., 2017). This approach helps maintain fairness and consistency in the dataset, making it more suitable for model training and evaluation.

Following previous work (Fatahi Bayat et al., 2023), the F1 metric is employed to assess the model's performance for automated and manual assessment, providing a comprehensive measure of precision and recall in error handling. For automated evaluation, GPT-4o (OpenAI, 2023) is selected as the judgment model. According to manual evaluation, the evaluator randomly sampled 10% of the data to conduct the assessment.

5.1.3 Implementation Details

All models are evaluated under zero-shot, 1-shot, 3-shot, CoT, and SFT settings with the corresponding instructions shown in Figures 13 to 15.

Before evaluation, our work conducts SFT on the LoRA (Hu et al., 2021) on several open-source models (LLaMA-3.2-3B, LLaMA-3.1-8B, Qwen-2.5-7B, Qwen-2.5-32B, Yi-6B, and Yi-34B). Training is conducted using LLaMA-Factory (Zheng et al., 2024b). The training process is conducted for three epochs, and the batch size is set to 2. The

Primary Category	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
Language Errors	6.50	7.19	3.93	17.95	13.34	20.23	11.53
Incomplete Information	40.58	42.27	41.32	14.78	7.83	**48.43**	32.53
Factual Errors	**72.99**	**74.51**	**71.70**	**48.22**	**41.63**	22.27	**55.22**
Logical Errors	41.49	54.39	43.04	44.29	30.91	18.36	38.74

Table 4: F1 Score of GPT-4o (OpenAI, 2023) in the four primary categories. Results are reported in percentage (%). "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr.", and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance.

learning rate is set to 1.0e-4, and its schedule follows a cosine curve. In the evaluation phase, for the zero-shot and SFT configurations, the input consists exclusively of the prompt without any supplementary instructions. In contrast, the 1-shot and 3-shot configurations include 1 and 3 exemplars, respectively. The temperature parameter is set to 0 to minimize randomness. Additionally, all models are instructed to output their responses in JSON format to facilitate statistical analysis.

5.1.4 Computing Platform

Our experiments are conducted on the workstation with an Intel Xeon E5 2.40 GHz CPU, four NVIDIA A800 GPUs, and CentOS 7.2.

5.2 Automated Experiment Result

Experiments are conducted to address the following research questions:
Question 1: How do LLMs perform on the four proactive error-handling tasks?
Question 2: Which errors are LLMs lacking in handling effectively?
Question 3: How can LLMs improve proactive error-handling capabilities?

Three answers are provided according to the above three questions.

5.2.1 LLM Performance Analysis

Overall Performance Table 3 presents the average results of each language model (LLM) across four tasks: detection, identification, correction, and guidance. Overall, closed-source models outperform their open-source counterparts, with Claude-3.5 exhibiting the highest average F1 of 50.83%, surpassing GPT-4o (OpenAI, 2023) (36.96%). This can be attributed to GPT-4o's tendency to focus more on directly answering the user's question rather than effectively identifying or correcting errors. The performance of LLaMA and Yi models aligns with the scaling law (Kaplan et al., 2020), where larger models generally achieve better performance. For instance, LLaMA-3.3 70B achieves an average performance of 46.29%, outperforming its smaller counterparts, LLaMA-3.2 3B (34.38%) and LLaMA-3.1 8B (35.15%). In contrast, Qwen-2.5 32B (43.24%) performs better than its larger counterpart, Qwen-2.5 72B (41.52%). This suggests the possible influence of inverse scaling (McKenzie et al., 2023), where increasing model size does not always guarantee improved performance and, in some cases, may even lead to performance degradation.

Comparison of Tasks The detection task is the simplest and achieves the highest F1 performance across all models, achieving 47.01%. This is likely because detecting the presence of errors is a relatively straightforward task compared to others. Attempting to identify at 50.34% generally exhibits higher performance than identifying correctly at 44.98%. This is because the model first attempts identification before successfully making the correct identification. The higher score in attempting to identify suggests that while models are capable of making identification attempts, they encounter difficulties in ensuring correctness, which demands more sophisticated processing. The Correction task proves to be more challenging, with overall performance at 30.86%, inferior to the Identification tasks. This is because the model must first be able to identify the error before it can attempt to correct it. Among all tasks, accurately correcting performs the worst at 22.62%, as it requires not only identifying the error correctly but also attempting to fix it before achieving accurate correction. This process may require richer prior knowledge and more nuanced reasoning. Finally, the performance of the Guidance task is relatively low compared to the other three tasks, at 29.41%, indicating that the model's ability to provide helpful guidance is the weakest. This task requires more advanced reasoning and the ability to navigate complex conversational prompts, making it particularly challenging for models with limited capabilities or reasoning.

For the first question, the answer is that **current**

Model	Setting	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
LLaMA-3.1-8B (Meta, 2024)	zero-shot	42.05	45.47	40.48	29.46	19.70	33.76	35.15
	1-shot	73.25	73.23	68.29	67.95	39.96	72.74	65.90
	3-shot	81.99	81.39	69.09	77.25	40.72	82.43	72.15
	CoT	75.62	77.00	73.56	72.65	47.02	75.44	70.22
	SFT	**90.16**	**90.59**	**80.02**	**82.20**	**62.86**	**84.77**	**81.77**
Qwen-2.5-32B (Qwen et al., 2024)	zero-shot	51.11	54.91	50.63	34.20	27.21	41.39	43.24
	1-shot	70.34	72.22	64.31	57.91	44.05	67.38	62.70
	3-shot	73.03	75.38	67.82	63.18	48.10	70.63	66.36
	CoT	72.90	77.21	74.75	72.44	58.15	80.02	72.58
	SFT	**97.88**	**97.98**	**88.43**	**88.96**	**70.86**	**93.17**	**89.55**
Yi-1.5-34B (01.AI et al., 2024)	zero-shot	46.40	48.25	42.41	31.39	22.36	10.63	33.57
	1-shot	68.29	70.07	64.65	57.17	38.66	52.19	58.51
	3-shot	75.23	76.18	66.73	61.49	44.73	56.61	63.50
	CoT	74.16	79.01	69.62	68.67	48.24	72.78	68.75
	SFT	**97.82**	**97.95**	**87.60**	**89.22**	**70.80**	**91.45**	**89.14**

Table 5: F1 Score comparison of different settings. Results are reported in percentage (%). "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr.", and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance.

LLMs lack sufficient proactive error-handling capabilities.

5.2.2 Error Category Analysis

Table 4 shows the F1 scores of GPT-4o (OpenAI, 2023) across various tasks, grouped by primary error categories. The model performs best on Factual Errors, with an F1 score of 0.5522, likely due to the extensive knowledge embedded in LLMs. It is followed by Logical Errors, with an F1 score of 0.3874. However, it struggles with Language Errors and Incomplete Information, with F1 scores of 0.1153 and 0.3253, respectively. This is because GPT-4o (OpenAI, 2023) tends to overlook these categories of errors and respond directly to them. Furthermore, our experimental results show that after SFT with Mis-prompt, LLMs' ability to proactively handle errors improves significantly, as shown in Table 5. A more fine-grained analysis of the secondary classification can be found in Appendix C.4.2.

For the second question, the answer is that **current LLMs fail to handle language errors, incomplete information, and logical errors**.

5.2.3 Technique Analysis

Table 5 presents the F1 scores of different language models (LLMs) across multiple tasks, comparing the performance of each model under various methods, including zero-shot, 1-shot, 3-shot, CoT, and SFT. SFT achieves the highest scores across all models. The 1-shot and 3-shot methods also show significant improvements over the zero-shot method but do not match the performance seen with CoT or SFT. For example, the LLaMA-3.2-8B model achieves an F1 score of 0.8177 under SFT, a significant improvement over its zero-shot score of 0.3515. The other methods score around 0.7, which is notably better than the zero-shot method but still inferior to SFT. A complete table of other models and their methods can be found in the Appendix C.

For the third question, the answer is that **SFT on error-handling instances is an effective method to improve LLMs' proactive error-handling capabilities**.

5.3 Human Evaluation Result

Table 12 shows the human evaluation results on the test set of Mis-prompt. Three master students are hired to conduct this work, and they are trained with the error categories in advance. Each sample is evaluated twice. The Fleiss' Kappa value of inter-annotator agreement is 0.63, showing strong consistency. If the evaluation results are consistent, that result is adopted. If the results are inconsistent, a language expert makes the final decision. The average discrepancy compared to the automated evaluation is 5.59%. The manual evaluation results align with those from automated assessment, demonstrating that SFT effectively enhances model capabilities in proactive error handling. Through CoT prompting and few-shot learning improvements over zero-shot approaches, their performance gains remain substantially inferior to those achieved through SFT in proactive error-handling tasks.

6 Conclusion

This work first identified a new challenge, which was proactive error handling. To promote further research on proactive error handling, this work introduced a novel evaluation framework comprising four key components: detection, identification, correction, and guidance. To support this study, this work developed Mis-prompt, a comprehensive benchmark that encompasses four main categories and 14 secondary categories of erroneous inputs. Experiment results show that current LLMs lack sufficient proactive error-handling capabilities, and SFT is an effective way to improve LLM performance in proactive error handling.

Limitations

First, our study primarily focuses on dialogue content consisting of pure text, and future work could investigate multimodal interactions that involve images, audio, or other forms of data. Additionally, our research has been limited to single-turn dialogues, which are relatively straightforward compared to multi-turn conversations. Furthermore, the F1 metric is utilized for scalable benchmarking, which ensures reproducibility but may not fully capture comprehensive aspects.

Ethical Statement

We make sure that Mis-prompt is collected in a manner that is consistent with the terms of use of any sources and the intellectual property and privacy rights of the original authors of the texts. And the crowd workers were treated fairly. This includes, but is not limited to, compensating them fairly, ensuring that they were able to give informed consent, and ensuring that they were voluntary participants who were aware of any risks of harm associated with their participation.

Acknowledgments

We would like to thank the School of Computer Science and Technology and the Institute of AI Education at East China Normal University for providing a computational platform. We also thank the reviewers for their insightful comments.

References

01.AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yanpeng Li, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Asma Ben Abacha, Wen-wai Yim, Yujuan Fu, Zhaoyi Sun, Meliha Yetisgen, Fei Xia, and Thomas Lin. 2024. Medec: A benchmark for medical error detection and correction in clinical notes. *arXiv preprint arXiv:2412.19260*.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

DeepSeek. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Farima Fatahi Bayat, Kun Qian, Benjamin Han, Yisi Sang, Anton Belyy, Samira Khorshidi, Fei Wu, Ihab Ilyas, and Yunyao Li. 2023. FLEEK: Factual error detection and correction with evidence retrieved from external knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 124–130, Singapore. Association for Computational Linguistics.

Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical error correction in low error density domains: A new benchmark and analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478.

Team GLM. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Gemini Team Google, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Ryo Kamoi, Sarkar Snigdha Sarathi Das, Renze Lou, Jihyun Janice Ahn, Yilun Zhao, Xiaoxin Lu, Nan Zhang, Yusen Zhang, Ranran Haoran Zhang, Sujeeth Reddy Vummanthala, Salika Dave, Shaobo Qin, Arman Cohan, Wenpeng Yin, and Rui Zhang. 2024. Evaluating llms at detecting errors in llm responses. *ArXiv*, abs/2404.03602.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander Richard Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. Summedits: measuring llm ability at factual reasoning through the lens of summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9662–9676.

Ruosen Li, Ziming Luo, and Xinya Du. 2024a. Fine-grained hallucination detection and mitigation in language model mathematical reasoning. *arXiv preprint arXiv:2410.06304*.

Xiaoyuan Li, Wenjie Wang, Moxin Li, Junrong Guo, Yang Zhang, and Fuli Feng. 2024b. Evaluating mathematical reasoning of large language models: A focus on error identification and correction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11316–11360, Bangkok, Thailand. Association for Computational Linguistics.

Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards conversational recommendation over multi-type dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1036–1049, Online. Association for Computational Linguistics.

Corina Masanti, Hans-Friedrich Witschel, and Kaspar Riesen. 2023. Novel benchmark data set for automatic error detection and correction. In *International Conference on Applications of Natural Language to Information Systems*, pages 511–521. Springer.

Ian R McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, et al. 2023. Inverse scaling: When bigger isn't better. *arXiv preprint arXiv:2306.09479*.

Meta. 2024. Llama 3 model card.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Zheng Ning, Yuan Tian, Zheng Zhang, Tianyi Zhang, and Toby Jia-Jun Li. 2024. Insights into natural language database query errors: From attention misalignment to user handling strategies. *ACM Transactions on Interactive Intelligent Systems*.

OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:13.

Yury Orlovskiy, Camille Thibault, Anne Imouza, Jean-François Godbout, Reihaneh Rabbany, and Kellin Pelrine. 2024. Uncertainty resolution in misinformation detection. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 93–101.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829.

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. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Madhvi Soni and Jitendra Singh Thakur. 2018. A systematic review of automated grammar checking in english language. *arXiv preprint arXiv:1804.00540*.

Zhivar Sourati, Vishnu Priya Prasanna Venkatesh, Darshan Deshpande, Himanshu Rawlani, Filip Ilievski, Hông-Ân Sandlin, and Alain Mermoud. 2023. Robust and explainable identification of logical fallacies in natural language arguments. *Knowledge-Based Systems*, 266:110418.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019a. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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 4149–4158.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019b. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Gladys Tyen, Hassan Mansoor, Peter Chen, Tony Mak, and Victor Cărbune. 2023. Llms cannot find reasoning errors, but can correct them! *arXiv preprint arXiv:2311.08516*.

Yibo Yan, Shen Wang, Jiahao Huo, Hang Li, Boyan Li, Jiamin Su, Xiong Gao, Yi-Fan Zhang, Tianlong Xu, Zhendong Chu, et al. 2024. Errorradar: Benchmarking complex mathematical reasoning of multimodal large language models via error detection. *arXiv preprint arXiv:2410.04509*.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2024a. Processbench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024b. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A Detailed Error Category Definition and Examples

A.1 Error Category Definition

Based upon previous research that defines error categories (Pagnoni et al., 2021; Sourati et al., 2023; Orlovskiy et al., 2024; Masanti et al., 2023), four distinct and prevalent primary error categories are identified: Language Errors, Incomplete Information, Factual Errors, and Logical Errors.

Language Errors Language errors refer to mistakes in the use of words, structures, or conventions when posing a question. These errors could be related to spelling, grammar, or punctuation, which hinder the clarity and accuracy of the question.

Spelling Errors Errors occur when words are incorrectly spelled (Masanti et al., 2023).

Grammatical Errors Errors happen when the structure or syntax of the question violates established rules of grammar (Masanti et al., 2023).

Punctuation Errors Errors occur when punctuation marks are misused or when necessary punctuation is missing (Masanti et al., 2023).

Incomplete Information The question lacks necessary background information, making it difficult to provide an accurate or meaningful answer. The question fails to provide sufficient context for the responder to understand or answer appropriately.

Speaker or Person The question involves a specific speaker or person but does not clarify who the speaker or person is or lacks the necessary context to understand the reference (Orlovskiy et al., 2024).

Textual Context and Subject Specification The question lacks specific context or details about the subject being referred to, leading to confusion or multiple possible interpretations (Orlovskiy et al., 2024).

Date and Time Period The question involves time or dates but does not specify the exact time period or date, leading to confusion or ambiguity (Orlovskiy et al., 2024).

Location When the question involves a location, it is not clear which location is being referred to, making it impossible to answer correctly (Orlovskiy et al., 2024).

Factual Errors The question is based on incorrect facts or assumptions, and the content mentioned does not align with reality.

Relation Error The relationship between entities or elements is incorrect or inconsistent (Pagnoni et al., 2021).

Entity Error The entities of the subject or object are incorrect (Pagnoni et al., 2021).

Circumstance Error Additional information outside the subject, predicate, and object (such as location or time) is incorrect. A logical fallacy in questioning is an error in reasoning that occurs when the logic behind a question is flawed, leading to an invalid or misleading question. This typically happens when the question contains hidden assumptions or faulty connections between concepts (Pagnoni et al., 2021).

Logical Errors A logical fallacy in questioning is an error in reasoning that occurs when the logic behind a question is flawed, leading to an invalid or misleading question. This typically happens when the question contains hidden assumptions or faulty connections between concepts.

Fallacy of Relevance The fallacy of Relevance occurs for arguments with premises that are logically irrelevant to the conclusion (Sourati et al., 2023).

Fallacy of Defective Induction Within the broad class of Fallacy of Defective Induction, the premises seemingly provide ground for the conclusion, but upon analysis, they prove to be insufficient and weak for supporting the claim made (Sourati et al., 2023).

Fallacy of Presumption The fallacy of Presumption occurs when the inference to the conclusion depends mistakenly on unwarranted assumptions (Sourati et al., 2023).

Fallacy of Ambiguity The fallacy of Ambiguity occurs when words or phrases are used in an equivocal way, thus causing ambiguity in the logic that connects the premise and the conclusion (Sourati et al., 2023).

A.2 Error Category Examples

This section provides illustrative examples of various error categories, corresponding prompts, and ground-truth responses.

Spelling Errors

- Prompt: Whear can I find the histori of the renowns Renaissance art?
- Ground truth: Your question contains spelling errors. The words “whear”, “histori”, and “renowns” are misspelled. They should be “where”, “history”, and “renowned”, respectively. The correct question should be: “Where can I find the history of the renowned Renaissance art?”

Grammatical Errors

- Prompt: What was the impact of the invention engine on industry how?
- Ground truth: Your question contains a grammatical error in word order. The specific error is the incorrect placement of the word "how'. The correct information is that "how" should appear before the noun phrase. You should phrase your question as: "How did the invention of the engine impact industry?"

Punctuation Errors

- Prompt: Which are the reasons Beethoven is considered a crucial figure; in the transition between the classical and romantic eras?
- Ground truth: Your question contains a punctuation error. The semicolon is incorrectly placed in the middle of a single clause, which disrupts the sentence flow. The semicolon should be replaced with a comma or removed entirely. The correct phrasing of your question should be: "What are the reasons Beethoven is considered a crucial figure in the transition between the classical and romantic eras?"

Speaker or Person

- Prompt: During the meeting, what did they argue about concerning the budget allocation?
- Ground truth: Your question contains an error due to the use of "they" without specifying who is being referred to. This makes it unclear which individuals or groups were involved in the argument. To correct this, please specify the individuals or groups you are asking about. For example, you could ask, "During the meeting, what did the finance team and the project managers argue about concerning the budget allocation?"

Textual Context and Subject Specification

- Prompt: Why did the system fail to increase output?
- Ground truth: Your question contains an error due to incomplete information. Specifically, it does not specify which system you are referring to, nor does it provide context regarding the criteria or conditions under which the output was supposed to increase. To get a more accurate response, please specify the system in question and provide additional context or conditions related to the expected increase in output. You might rephrase your question as: "Why did the [specific system] fail to increase output under [specific conditions]?"

Date and Time Period

- Prompt: Bobby thought Bill should buy a trailer and haul it with his car. Bill thought a truck would be better for what he needed. Bobby pointed out two vehicles were much more expensive. Bill was set in his ways with conventional thinking. He ended up buying the truck he wanted despite Bobby's advice. When did Bill purchase the truck?
- Ground truth: The user's question contains an error because it asks for a specific time or date when Bill purchased the truck, which is not provided in the story. The story does not mention when the purchase took place. To phrase the correct question, you might ask, "What decision did Bill make regarding the vehicle purchase?"

Location

- Prompt: Melody's parents surprised her with a trip to the big aquarium. Melody took a nap during the two-hour car ride to the aquarium. When they arrived, Melody was another excited animal. At the aquarium, Melody saw sharks, tropical fish, and many other animals. After five hours at the aquarium, Melody and her family drove home. What city is the aquarium located in?
- Ground truth: The question contains an error because it asks for the city where the aquarium is located, but the story does not provide this information. To obtain the correct information, you would need to know the specific city mentioned in the context. You can phrase the correct question by asking, "Can you provide more details about the location of the aquarium mentioned in the story?"

Relation Error

- Prompt: How did Thomas Edison win the Nobel Prize for Literature with his invention of the phonograph?

- Ground truth: Your question contains an error. Thomas Edison did not win the Nobel Prize for Literature, nor did he receive any Nobel Prize for his inventions. Edison was an inventor known for the phonograph, light bulb, and other devices, but the Nobel Prizes do not typically recognize inventions under the category of literature. To phrase the correct question, you might ask: "What are some of Thomas Edison's notable inventions?"

Entity Error

- Prompt: Which Shakespeare play ends with the character Frodo Baggins casting the One Ring into Mount Doom?

- Ground truth: Your question contains an error. Frodo Baggins is a character from "The Lord of the Rings," a novel by J.R.R. Tolkien, not a Shakespeare play. To ask a correct question, you might say: "Which Shakespeare play ends with a significant event or resolution?"

Circumstance Error

- Prompt: When did the Wright brothers make their first flight in New York City in 1920?

- Ground truth: Your question contains an error. The Wright brothers did not make their first flight to New York City in 1920. Their first successful flight took place in 1903 in Kitty Hawk, North Carolina. To ask the correct question, you could phrase it as: "When and where did the Wright brothers make their first successful flight?"

Fallacy of Relevance

- Prompt: Considering that apples originate from Kazakhstan, how does growing oranges in Florida affect the climate policies of European countries?

- Ground truth: Your question contains a logical error known as the fallacy of relevance. The origin of apples in Kazakhstan is unrelated to the cultivation of oranges in Florida and the climate policies of European countries. To address the topic correctly, you might ask: "How does the cultivation of oranges in Florida impact local climate policies, and what lessons can European countries learn from these practices?"

Fallacy of Defective Induction

- Prompt: Given that a few celebrities endorse the ketogenic diet, how does this prove it is the healthiest option for everyone?

- Ground truth: Your question contains an error known as the fallacy of defective induction. The error lies in assuming that endorsements from a few celebrities are sufficient to prove that the ketogenic diet is the healthiest option for everyone. This overlooks the diverse dietary needs and health conditions of individuals. A more accurate question would be: "What are the health benefits and potential risks of the ketogenic diet, and how can individuals determine if it is suitable for their specific dietary needs?"

Fallacy of Presumption

- Prompt: Since all dogs can fly, how should we construct kennels to accommodate their flights?

- Ground truth: Your question contains an error due to a presumption without basis in reality. Specifically, the error is the assumption that all dogs can fly, which is not true. Dogs do not have the ability to fly. To phrase the correct question, you might ask: "How should we construct kennels to accommodate the needs of dogs?"

Fallacy of Ambiguity

- Prompt: Since we know that time flies when you're having fun, how does it affect the wings" strength?

- Ground truth: Your question contains an error due to a misunderstanding of the idiom "time flies." The phrase "time flies" is an idiomatic expression meaning that time seems to pass quickly, especially when one is enjoying themselves. It does not imply that time is a tangible entity with wings. To phrase your question correctly, you might ask: "Since time seems to pass quickly when you're having fun, what are some ways to make the most of enjoyable moments?"

B Generation Rules Design

Generation rules are formulated for each category to generate cases that conform to the defined error categories.

Spelling Errors Error To create a spelling error, select some words in the sentence and deliberately misspell them. The incorrect spelling should confuse and be wrong to the reader.

Grammatical Errors Introduce a syntactical error in the sentence structure, such as improper subject-verb agreement, incorrect word order, or missing auxiliary verbs. This modification will result in a grammatical error, causing the sentence to be faulty.

Punctuation Errors To introduce a punctuation error, either omit necessary punctuation marks or place them incorrectly within the question. The mistake should confuse the meaning. Please do not limit the errors to just missing question marks(?) and periods(.), but focus on punctuation errors that have a more significant impact on the sentence's meaning.

Speaker or Person Omit the specific entity involved in the question (e.g., person, object, or concept). This creates a severe gap in the question's context, leaving the responder without any clue about what is being asked, which makes it almost impossible to provide a relevant answer.

Location The question should refer to a specific location but lack the necessary details (e.g., city, country, or a well-known landmark) that would allow a meaningful or accurate answer. The omission of these details should render the question ambiguous, making it impossible to provide a precise response. Do not ask questions that explicitly omit location names like "where"' or "which place'—the key is in the context where the missing information leaves the question unanswerable or unclear. The question should appear normal, but the absence of a crucial detail (like the exact name of a place) should create a scenario where the answer cannot be determined without further clarification.

Textual Context and Subject Specification Omit specific details related to the subject or object of the question (e.g., key attributes, conditions, or essential components of the query). This omission leaves the question incomplete and ambiguous, making it difficult for the responder to identify what is being asked accurately. The missing information should not include the speaker, person, location, or time/period, as those are fundamental for context, but should focus on other critical details that are necessary for a precise and relevant answer.

Relation Error Alter the predicate (the main assertion or claim in the question) to contradict the source of information. This creates a relationship error, where the relationship between two concepts or facts is incorrectly stated, misleading the reader or making the question factually inaccurate.

Entity Error Change or confuse the subject of the question, such as the name of an entity or its specific characteristic. The question becomes factually incorrect, as it attributes the wrong identity or characteristic to the entity, leading to a misleading or erroneous conclusion.

Circumstance Error Alter the additional details related to the context, such as time, place, or specific conditions. This could involve changing the circumstances (e.g., date, location, or situation) surrounding the main action, which results in an incorrect interpretation of the event or subject.

Fallacy of Relevance Introduce an irrelevant premise that distracts from the core issue or conclusion. This leads to a question that logically misleads or confuses the responder by focusing on an unrelated or unimportant issue.

Fallacy of Defective Induction To generate the Fallacy of Defective Induction, start with a strong and well-supported argument, then introduce either insufficient or weak evidence, such as a small sample size or unrepresentative data. The evidence should still seem to support the conclusion, but it should be inadequate to reach such a broad or strong claim justifiably. The key error is in assuming that weak or limited evidence can logically lead to a general conclusion, creating a misleading argument that appears convincing but lacks proper support.

Fallacy of Presumption For the Fallacy of Presumption, begin with a correct argument or claim and introduce an assumption that is taken for granted without evidence. This assumption should be treated as self-evident, even though it is unproven or unjustified. The error here lies in building an argument on this untested assumption as if it is a fact, leading to a flawed reasoning process. The mistake is presuming something to be true without providing any supporting evidence, causing the argument to be based on an unverified premise.

Fallacy of Ambiguity Use terms or phrasing that are inherently ambiguous or have multiple, conflicting meanings so that the question can be interpreted in various ways. This increases the likelihood of the respondent misunderstanding or providing an irrelevant answer, as the question will not only be unclear but will actively lead to divergent interpre-

tations, making it nearly impossible to determine what is truly being asked.

C Experiment Details

This section provides a comprehensive overview of the prompts, input formatting, and specific experimental results.

C.1 Error-Handling Task Instructions

This section provides detailed instructions and methodologies for the Error-Handling Task, outlining the design and implementation of prompts and evaluation protocols.

C.1.1 Task 1: Error Detection

Figure 4 illustrates the prompt design and input format for the error detection task, which aims to detect whether there are errors in the model's response to user input using the judgment model.

C.1.2 Task 2: Error Identification

Figure 5 presents the prompt and input structure for the error identification task, which aims to use a judgment model to determine whether the response attempts and correctly identifies the specific category and location of errors.

C.1.3 Task 3: Error Correction

Figure 6 outlines the prompt strategy for the error correction task, where the objective is to use the judgment model to assess whether the response attempts and correctly corrects the errors.

C.1.4 Task 4: Error Guidance

Figure 7 displays the prompt and format design for the error guidance task, which leverages the judgment model to evaluate whether the response provides users with actionable advice to prevent similar errors in the future.

C.2 Dataset Generation

Data generation follows two approaches. One approach involves transforming existing datasets with the designed prompts shown in Figures 8 to 10. The other approach directly generates the dataset, with the prompt illustrated in Figure 11. Finally, ground-truth responses are generated using the prompts from Figure 12. These prompts are carefully designed to ensure the diversity and quality of the dataset, covering a wide range of scenarios and error categories.

C.3 Evaluation Models' Response Generation

The tested models have the following five settings: zero-shot, 1-shot, 3-shot, CoT, and SFT. In the zero-shot and SFT setups, the input is limited to the prompt itself, with no additional instructions provided. The instruction templates for 1-shot, 3-shot, and CoT are displayed in Figures 13, 14, and 15, respectively.

C.4 Detailed results

In this section, we present the original detailed experimental results.

C.4.1 Main Experiment

Table 6 presents the overall performance of the models. It could be observed that Closed-source models, especially Claude-3.5, exhibit superior performance. Task difficulty increases from detection to identification, correction, and guidance, highlighting the need for improved reasoning and knowledge utilization in more complex tasks. Furthermore, SFT significantly enhances the model's error-handling capabilities. While few-shot learning and CoT also demonstrate improvements, their performance is comparable to each other and notably inferior to that of SFT.

C.4.2 Error Category Analysis

The performance of different baselines across various primary error categories is detailed in Tables 7 to 10.

A deeper look at the secondary categories reveals varied performance across error categories, as shown in Table 11. These results highlight the model's strengths and weaknesses across different secondary categories, demonstrating notable variability in how it handles various categories of errors.

Language Errors category In the Language Errors category, grammatical errors stand out with strong performance across detection, attempt identification, and guidance metrics. This suggests that the model handles grammatical errors more effectively compared to punctuation and spelling errors.

Incomplete Information For Incomplete Information, the model performs exceptionally well in identifying the speaker or person. In contrast, location-related errors are more challenging for the model, showing lower scores. The model also performs well with Date and Time Period errors, indicating strong handling of these specific categories of information gaps.

Factual Errors In the Factual Errors category, entity errors lead to overall performance, with the model demonstrating a high level of score in recognizing and handling entity-related errors. On the other hand, Relation Errors show relatively strong performance, while Circumstance Errors present moderate performance across the board.

Logical Errors Lastly, in the Logical Errors category, the Fallacy of Defective Induction stands out with the highest scores across all metrics, suggesting that the model handles these errors well. The Fallacy of Presumption and the Fallacy of Relevance follow closely behind. However, the Fallacy of Ambiguity has the lowest performance, indicating that the model struggles with identifying ambiguous logical errors effectively.

C.4.3 Human Evaluation

We randomly sampled 10% of the evaluation set for manual evaluation. The results are presented in Table 12.

Instruction for Error Detection

 # Evaluation Tasks for Error Handling in Prompts
Task 1: Error Detection
- **Goal:** Determine whether the `response` detects an error in the `prompt`.
- **Criteria:**
  - *** detection:** If the `response` explicitly states that the prompt contains an error (e.g., "Your question has an error", "There is a mistake", "This is incorrect", or similar expressions), output `"yes"`. If the `response` does not mention an error at all, output `"no"`.

Format Requirement:
```
{
  "detection": "<yes/no>"
}
```

Output Only:
Your response should be in the specified JSON format, without additional text or explanation outside the JSON structure.

Input:
prompt
{prompt}
response
{response}
gold answer
{gold_answer}

Figure 4: Instruction for *Error Detection*.

Instruction for Error Identification

 # Evaluation Tasks for Error Handling in Prompts
Task 2: Error Identification
- **Goal**: Determine whether the `response` attempts to identify the error and whether it correctly identifies it.
- **Criteria:**
  - *** attempted:** If the `response` includes a specific description of the error type or reason (e.g., "due to a logical fallacy", "syntax error", "invalid assumption"), output `"yes"`. If the reason is not specified (e.g., "This is wrong"), output `"no"`.
  - *** correct:** If the error reason described in the `response` aligns with the description in the `gold_answer` (logical relationship is entailment), output `"yes"`. Otherwise, output `"no"` (logical relationship is neutral or contradictory).

Format Requirement:
```
{
  "identification": {
    "attempted": "<yes/no>",
    "correct": "<yes/no>"
  }
}
```

Output Only:
Your response should be in the specified JSON format, without additional text or explanation outside the JSON structure.

Input:
prompt
{prompt}
response
{response}
gold answer
{gold_answer}

Figure 5: Instruction for *Error Identification*.

Instruction for Error Correction

 # Evaluation Tasks for Error Handling in Prompts
Task 3: Error Correction
- **Goal:** Determine whether the `response` attempts to correct the error and whether the correction is correct.
- **Criteria:**
  - *** attempted:** If the `response` provides a correction, output `"yes"`. If no correction is provided (e.g., "This is invalid"), output `"no"`.
  - *** correct:** If the correction in the `response` aligns with the description in the `gold_answer` (logical relationship is entailment), output `"yes"`. Otherwise, output `"no"` (logical relationship is neutral or contradictory).

Format Requirement:
```
{
  "correction": {
    "attempted": "<yes/no>",
    "correct": "<yes/no>"
  },
}
```

Output Only:
Your response should be in the specified JSON format, without additional text or explanation outside the JSON structure.

Input:
prompt
{prompt}
response
{response}
gold answer
{gold_answer}

Figure 6: Instruction for *Error Correction*.

Instruction for Error Guidance

 # Evaluation Tasks for Error Handling in Prompts
Task 4: Error Guidance
- **Goal:** Determine whether the `response` guides the user to improve the question.
- **Criteria:**
    * **guidance:** If the `response` includes explicit guiding statements (e.g., "Please clarify your question", "Try asking...", "You might rephrase it as..."), output `"yes"`. If it does not guide the user, output `"no"`.

Format Requirement:
```
{
   "guidance": "<yes/no>"
}
```

Output Only:
Your response should be in the specified JSON format, without additional text or explanation outside the JSON structure.

Input:
prompt
{prompt}
response
{response}
gold answer
{gold_answer}

Figure 7: Instruction for *Error Guidance*.

Instruction for Converting FEVEROUS into Mis-prompt

Task Overview:
You are tasked with generating structured cases based on the provided refuted claims and evidence.
Each case should include the error category, an erroneous user query (Prompt), and a supporting explanation.

Error Category:
To help you classify errors, here are the primary and secondary categories:
[Factual Errors]
Relation Error: The relationship between entities or elements is incorrect or inconsistent.
Entity Error: The entities of the subject or object are incorrect.
Circumstance Error: Additional information outside the subject, predicate, and object (such as location or time) is erroneous.

Steps for Case Creation: Given the provided refuted claims and evidence, follow these steps:
1. Identify the error type and classify it under both a primary and secondary category. Please ensure that the categorization is correct and not ambiguous.
2. Ensure the error appears in the following additional guidelines while avoiding language errors.
3. Provide an accurate and concise explanation of the error, including why the error belongs to that particular category and the reason behind it. The explanation does not need to mention the claim and evidence, only stating the facts.

Additional Guidelines:
* Errors in question structure: 1)Special questions with errors: These questions contain an error in the main part of the question. 2) Incorrect information in subordinate clauses: This type of error occurs when the subordinate clause contains misleading or incorrect information, which can mislead the direction of the answer. 3) Incorrect statement + question: In this structure, the question begins with an incorrect statement, followed by a related question.
* The erroneous question should primarily be a special question rather than a general yes/no question. This means the error should relate to specific information requested, rather than a simple affirmation or negation.
* Ask a question that the original text cannot answer.

Format Requirement:
Please provide your response in JSON format as follows:
```
{
   "primary-category": "<primary-category>",
   "secondary-category": "<secondary-category>",
   "error-prompt": "<User erroneous question>",
   "explanation": "<Concise and accurate explanation>"
}
```

Example Cases:
case 1:
```
{
   "primary-category": "Factual Errors",
   "secondary-category": "Entity Error",
   "error-prompt": "During the 2006-07 season, the San Jose Sharks scored the most points in the Pacific Division. How did they achieve this?",
   "explanation": "This is because the claim incorrectly states that the San Jose Sharks scored the most points in the Pacific Division during the 2006-07 season, whereas the Anaheim Ducks actually scored more points."
}
```
case 2:
```
{
   "primary-category": "Factual Errors",
   "secondary-category": "Relation Error",
   "error-prompt": "Since Lower Darwen railway station is open for passengers currently, what are the operating hours?",
   "explanation": "This is because the claim incorrectly states that Lower Darwen railway station is open for passengers, when in fact it is classified as 'Disused' and is no longer in operation."
}
```
case 3:
```
{
   "primary-category": "Factual Errors",
   "secondary-category": "Entity Error",
   "error-prompt": "Ortegal has three municipalities with a total population of 12,238 in 2018, one of which was called Cerdido. What are the names of the other two municipalities?",
   "explanation": "This is because the claim incorrectly states that Ortegal has three municipalities, while the evidence shows there are four municipalities."
}
```
case 4:
```
{
   "primary-category": "Factual Errors",
   "secondary-category": "Relation Error",
   "error-prompt": "Stefan Kahil Lessard was born outside Anaheim, California. Where exactly was he born?",
   "explanation": "This is because the claim incorrectly identifies the birth location of Stefan Kahil Lessard as outside Anaheim, California, whereas he was actually born in Anaheim, California."
}
```
case 5:
```
{
   "primary-category": "Factual Errors",
   "secondary-category": "Circumstance Error",
   "error-prompt": "Geordie Henderson joined Rangers in November 1818 and was top scorer for four consecutive seasons. What were his achievements during this period?",
   "explanation": "This is because Geordie Henderson joined Rangers in November 1919, not 1818. The incorrect year creates a misleading historical context."
}
```

Output Only: Your response should be in the specified JSON format, without additional text or explanation outside the JSON structure.

Case Details:
* claim: {provided_claim}
* evidence: {provided_evidence}

Figure 8: Instruction for converting *FEVEROUS* into Mis-prompt.

Instruction for Converting CommonsenseQA into Mis-prompt

Task Overview:
You are tasked with generating structured cases based on the provided claim. Each case should include the error category, an erroneous user query (Prompt), and a supporting explanation.

Error Category:
To help you classify errors, here are the primary and secondary categories:
[Logical Errors]
Fallacy of Relevance: Fallacy of Relevance occurs for arguments with premises that are logically irrelevant to the conclusion,
Fallacy of Defective Induction: Within the broad class of Fallacy of Defective Induction, the premises seemingly provide ground for the conclusion, but upon analysis prove to be insufficient and weak for supporting the claim made,
Fallacy of Presumption: Fallacy of Presumption occurs when the inference to the conclusion depends mistakenly on unwarranted assumptions,
Fallacy of Ambiguity: Fallacy of Ambiguity occurs when words or phrases are used equitably, thus causing ambiguity in the logic that connects the premise and the conclusion.

Steps for Case Creation: Given the provided question and its wrong answer, follow these steps:
1. Identify the error type and classify it under both a primary and secondary category. Please ensure that the categorization is correct and not ambiguous.
2. Convert the claim into an error query, ensuring that the error appears according to the following additional guidelines while avoiding language errors.
3. Provide an accurate and concise explanation of the error, including why the error belongs to that particular category and the reason behind it.

Additional Guidelines:
* Errors in question structure: 1) Special questions with errors: These questions contain an error in the main part of the question. 2) Incorrect information in subordinate clauses: This type of error occurs when the subordinate clause contains misleading or incorrect information, which can mislead the direction of the answer. 3) Incorrect statement + question: In this structure, the question begins with an incorrect statement, followed by a related question.
* The erroneous question should primarily be a special question rather than a general yes/no question. This means the error should relate to specific information requested, rather than a simple affirmation or negation.
* Ask a question that the original text cannot answer.

Format Requirement:
Please provide your response in JSON format as follows:
```
{
  "primary-category": "<primary-category>",
  "secondary-category": "<secondary-category>",
  "error-prompt": "<User erroneous question>",
  "explanation": "<Concise and accurate explanation>"
}
```

Example Cases:
case 1:
```
{
  "primary-category": "Logical Errors",
  "secondary-category": "Fallacy of Relevance",
  "error-prompt": "Since the Earth is flat, how does it affect the gravitational pull?",
  "explanation": "This is a fallacy of relevance because the shape of the Earth is irrelevant to the explanation of gravitational pull. The premise that the Earth is flat is factually incorrect and has no bearing on the actual scientific principle of gravity."
}
```
case 2:
```
{
  "primary-category": "Logical Errors",
  "secondary-category": "Fallacy of Defective Induction",
  "error-prompt": "The last 5 companies I invested in went bankrupt. So why is investing in stocks always risky?",
  "explanation": "This is a fallacy of defective induction because the user's personal experience with a few companies does not provide sufficient evidence to claim that investing in stocks is always risky. The sample size is too small and not representative of the entire stock market."
}
```
case 3:
```
{
  "primary-category": "Logical Errors",
  "secondary-category": "Fallacy of Presumption",
  "error-prompt": "Since all politicians are corrupt, how can we trust any of their policies?",
  "explanation": "This is a fallacy of presumption because the claim that all politicians are corrupt is an unwarranted assumption. It assumes that all politicians share the same negative trait without proof, leading to an invalid conclusion."
}
```
case 4:
```
{
  "primary-category": "Logical Errors",
  "secondary-category": "Fallacy of Ambiguity",
  "error-prompt": "How does Python crawl? Why can it crawl data?",
  "explanation": "This is a fallacy of ambiguity because the word 'crawl' is being used in two different senses: one meaning the physical act of crawling (as an animal might do), and the other referring to web crawling (the process of collecting data from the internet using a program). The question does not clarify which sense of 'crawl' it refers to, leading to confusion about the intended meaning."
}
```
case 5:
```
{
  "primary-category": "Logical Errors",
  "secondary-category": "Fallacy of Relevance",
  "error-prompt": "She won the lottery. Why do people still think she deserves to be president?",
  "explanation": "This is a fallacy of relevance because winning the lottery has no logical connection to someone's qualifications to be president. The question introduces an irrelevant detail (winning the lottery) that does not address the person's political capabilities."
}
```

Output Only: Your response should be in the specified JSON format, without additional text or explanation outside the JSON structure.

Case Details:
* Chaim: {provided_chaim}

Figure 9: Instruction for converting *CommonsenseQA* into Mis-prompt.

Instruction for Converting ROCStories into Mis-prompt

Task Overview:
You are tasked with generating structured cases based on the provided story.
Each case should include the error category, an erroneous user query (Prompt), and a supporting explanation.

Steps for Case Creation: Given the provided story, follow these steps:
1. The question about {type} cannot be answered based on the original text.
2. Output the modified story with the question. The questions should exhibit diversity and strictly adhere to the missing type. Avoid allowing the "Textual context and subject specification" category to ask questions related to person, time, or location.
3. Provide an accurate and concise explanation of the error.

Additional Guidelines:
* Errors in question structure: 1)Special questions with errors: These questions contain an error in the main part of the question. 2) Incorrect information in subordinate clauses: This type of error occurs when the subordinate clause contains misleading or incorrect information, which can mislead the direction of the answer. 3) Incorrect statement + question: In this structure, the question begins with an incorrect statement, followed by a related question.
* The erroneous question should primarily be a special question rather than a general yes/no question. This means the error should relate to specific information requested, rather than a simple affirmation or negation.
* Ask a question that the original text cannot answer.

Format Requirement:
Please provide your response in JSON format as follows:
{
"primary-category": "<primary-category>",
"secondary-category": "<secondary-category>",
"error-prompt": "<User erroneous question>",
"explanation": "<Concise and accurate explanation>"
}

Example Case:

Case 1:
{
"primary-category": "Incomplete Information",
"secondary-category": "Speaker or person",
"error-prompt": "The man liked the flavor. He tried to recreate it at home. He could not get the flavor right. He asked the owner of the recipe for help. The owner of the flavor sold him the recipe. Who is the owner of the recipe? Has the owner won any awards?",
"explanation": "The story does not provide any information about the identity of the owner of the recipe or whether they have won any awards, leaving those details unclear."
}
Case 2:
{
"primary-category": "Incomplete Information",
"secondary-category": "Textual context and subject specification",
"error-prompt": "Jane only had one pair of glasses. And she had broken them. Her mother taped them up and sent her to school. When she entered the classroom everyone stared at her. They all pointed and laughed as she stood wishing she could disappear. How did Jane react emotionally when she entered the classroom and saw everyone's reaction?",
"explanation": "The story does not provide any information about Jane's emotional response to the situation in the classroom, leaving her feelings unknown."
}
Case 3:
{
"primary-category": "Incomplete Information",
"secondary-category": "Location",
"error-prompt": "Janice was out exercising for her big soccer game. She was doing some drills with her legs. While working out and exercising she slips on the grass. She falls down and uses her wrist to break her fall. She breaks her wrist in the process and goes to the hospital. What type of surface was Janice exercising on that led to her slip?",
"explanation": "The story does not specify the type of surface Janice was exercising on, leaving that detail about the location of the accident unclear."
}
Case 4:
{
"primary-category": "Incomplete Information",
"secondary-category": "Date and time period",
"error-prompt": "Jamie is an American girl. Jamie wants to get married to a Mexican man. Her family assumes it's because the man wants a green card. Jamie insists that she is marrying him out of love. Jamie gets married and they spend the rest of their lives together. What time of year did Jamie get married?",
"explanation": "The story does not mention when Jamie got married, leaving the time period of the marriage unclear."
}

Case 5:
{
"primary-category": "Incomplete Information",
"secondary-category": "Location",
"error-prompt": "The orange fell from the tree. It hit a girl on the head. The girl looked up at the tree. Another orange fell from the tree. That orange broke her nose. What kind of tree is this? Is it known for producing large fruit?",
"explanation": "The story does not provide any information about the type of tree or the characteristics of its fruit, leaving these details unclear."
}

Output Only: Your response should be in the specified JSON format, without additional text or explanation outside the JSON structure.

Case Details:
* story: {provided_story}

Figure 10: Instruction for converting *ROCStories* into Mis-prompt.

Instruction for Directly Generate Mis-prompt

Task Overview:
You are tasked with generating structured cases based on the error method design.
Each case should include the error category, an erroneous user query (Prompt), and a supporting explanation.

Steps for Case Creation: Given the error method design, follow these steps:
1. Generate the question according to the error method design, ensuring that the error appears as either an incorrect statement followed by a question or a question with a misleading clause
2. Provide an accurate and concise explanation of the error.

Additional Guidelines:
* Errors in question structure: 1) Special questions with errors: These questions contain an error in the main part of the question. 2) Incorrect information in subordinate clauses: This type of error occurs when the subordinate clause contains misleading or incorrect information, which can mislead the direction of the answer. 3) Incorrect statement + question: In this structure, the question begins with an incorrect statement, followed by a related question.
* Avoid using a standalone question. Try to express the idea in a more detailed and extended manner, with longer content.
* The erroneous question should primarily be a special question rather than a general yes/no question. This means the error should relate to specific information requested, rather than a simple affirmation or negation.
* Please ask a question that the original text cannot answer. As you generate your question, consider whether the question is answerable based on the content provided. If the question can be answered from the text, adjust it so that it cannot be answered directly.

Format Requirement:
Please provide your response in JSON format as follows:
```
{
   "primary-category": {primary_category},
   "secondary-category": {secondary_category},
   "error-prompt": "<User erroneous question>",
   "explanation": "<Concise and accurate explanation>"
}
```

Objective: {objective}
Generation Rules Design: {method_description} // Error Method Design

Output Only: Your response should be in the specified JSON format, without additional text or explanation outside the JSON structure.

Please generate {count} cases at once, ensuring there are significant differences between them. Output them in the following format: [{{}},{{}},{{}},{{}},{{}}]. Ensure that the output contains only this JSON array format, with no additional content.

Figure 11: Instruction for generating Mis-prompt directly.

Instruction for Generate Ground-Truth

Task Overview:
You are required to generate a structured response based on the provided error-prompt, category, and explanation.

Each response should:
1. Point out that the user's question contains an error.
2. Identify the specific error.
3. Provide the correct information or concept.
4. Guide the user on how to phrase the correct question.

Output Only: Provide your response without additional text or explanation.

Case Details:
* error-prompt: {error_prompt}
* explanation: {explanation}
* category: {primary-category}-{secondary-category}

Figure 12: Instruction for generating the ground truth.

Instruction for 1-shot Evaluation

Example 1:
Question: "How long does it take to drive from the university to the beach?"
Answer: "Your question contains an error due to incomplete information regarding the locations. Specifically, it does not specify which university or beach you are referring to. Travel time can vary significantly based on these details. To get an accurate answer, please specify the names of the university and the beach. For example, you could ask, 'How long does it take to drive from Stanford University to Santa Cruz Beach?'"

Now, please answer the following question:
Question: {prompt}

Figure 13: Instruction for 1-shot evaluation.

Instruction for 3-shot Evaluation

Example 1:
Question: "How long does it take to drive from the university to the beach?"
Answer: "Your question contains an error due to incomplete information regarding the locations. Specifically, it does not specify which university or beach you are referring to. Travel time can vary significantly based on these details. To get an accurate answer, please specify the names of the university and the beach. For example, you could ask, 'How long does it take to drive from Stanford University to Santa Cruz Beach?'"

Example 2:
Question: "What are the reasons for the variety of fuel types used in modern Russian vehicles?"
Answer: "The variety of fuel types used in modern Russian vehicles is influenced by factors such as the country's diverse climate and geography, economic considerations, infrastructure availability, government policies, environmental concerns, technological advancements, military and industrial needs, and consumer preferences. These elements together create a diverse fuel landscape that caters to different regional and consumer needs."

Example 3:
Question: "During the 2017-2018 season, the Liberty Lady Flames basketball team played their final season in the Atlantic Sun Conference. What were their achievements in this conference?"
Answer: "Your question contains an error regarding the conference in which the Liberty Lady Flames basketball team played during the 2017-2018 season. Specifically, they were in the Big South Conference, not the Atlantic Sun Conference, during that season. The Liberty Lady Flames moved to the Atlantic Sun Conference after the 2017-2018 season. To phrase the correct question, you could ask: 'What were the achievements of the Liberty Lady Flames basketball team during their final season in the Big South Conference in 2017-2018?'"

Now, please answer the following question:
Question: {prompt}

Figure 14: Instruction for 3-shot evaluation.

Instruction for CoT Evaluation

Let's think through this step by step: You will be given a question. First, check if there are any mistakes in the question. If the question is correct, provide the answer. If there is an error, first tell the user that there is an error in the question, identify what is wrong, and correct the question if necessary. Then, guide the user to rephrase the question.

Question: {prompt}

Figure 15: Instruction for CoT evaluation.

Model	Setting	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
GPT-4o (OpenAI, 2023)	zero-shot	43.54	48.71	43.78	31.72	23.32	30.66	36.96
	1-shot	**85.63**	**85.73**	73.92	76.74	47.88	**81.28**	75.20
	3-shot	82.70	84.34	75.65	74.39	54.55	75.48	74.52
	CoT	82.19	84.53	**80.71**	**80.86**	**60.75**	77.39	**77.74**
Gemini-1.5 (Google et al., 2024)	zero-shot	55.13	**60.73**	**54.67**	**28.38**	**22.21**	**23.56**	**40.78**
	1-shot	68.88	71.00	69.11	41.67	29.59	38.58	53.14
	3-shot	71.31	**74.34**	70.65	45.93	33.37	45.58	56.86
	CoT	**72.93**	73.97	**77.44**	**72.21**	**62.83**	**73.92**	**72.22**
Claude-3.5 (Anthropic, 2024)	zero-shot	63.98	67.53	63.01	36.48	30.23	43.73	50.83
	1-shot	**88.19**	**88.17**	**80.54**	**79.15**	**58.64**	**81.03**	**79.29**
	3-shot	86.59	86.55	81.44	**79.82**	62.35	**82.33**	**79.85**
	CoT	83.24	83.75	**82.37**	77.18	**64.14**	81.17	78.64
GLM-4 (GLM, 2024)	zero-shot	53.18	56.19	50.40	37.19	28.63	32.04	42.94
	1-shot	81.22	82.06	75.10	76.08	55.48	81.31	75.21
	3-shot	**83.21**	**83.81**	**77.23**	**80.84**	**59.96**	**81.17**	**77.70**
	CoT	79.96	81.55	76.86	67.87	48.93	**82.09**	72.88
LLaMA-3.2-3B (Meta, 2024)	zero-shot	43.60	44.56	39.96	26.71	18.07	33.39	34.38
	1-shot	69.10	69.20	60.92	65.28	30.60	68.99	60.68
	3-shot	66.56	66.39	65.43	68.11	37.42	65.39	61.55
	CoT	72.20	75.15	63.24	68.61	36.48	74.91	65.10
	SFT	**97.48**	**97.74**	**82.98**	**88.02**	**63.69**	**91.99**	**86.98**
LLaMA-3.1-8B (Meta, 2024)	zero-shot	42.05	45.47	40.48	29.46	19.70	33.76	35.15
	1-shot	73.25	73.23	68.29	67.95	39.96	72.74	65.90
	3-shot	81.99	81.39	69.09	77.25	40.72	82.43	72.15
	CoT	75.62	77.00	73.56	72.65	47.02	75.44	70.22
	SFT	**90.16**	**90.59**	**80.02**	**82.20**	**62.86**	**84.77**	**81.77**
LLaMA-3.3-70B (Meta, 2024)	zero-shot	57.78	59.23	53.50	39.67	30.17	37.40	46.29
	1-shot	75.36	75.43	73.41	71.57	46.33	78.02	70.02
	3-shot	85.32	85.32	77.60	**79.65**	55.84	**85.08**	78.14
	CoT	**85.74**	**85.95**	**82.22**	74.91	**59.36**	84.29	**78.75**
Qwen-2.5-7B (Qwen et al., 2024)	zero-shot	43.88	47.59	43.07	31.47	22.95	37.71	37.78
	1-shot	58.84	61.43	53.67	44.66	30.18	58.74	51.25
	3-shot	69.63	70.44	60.24	59.46	40.37	66.42	61.09
	CoT	71.17	75.52	70.08	66.14	43.03	66.84	65.46
	SFT	**95.99**	**96.24**	**86.52**	**85.20**	**67.66**	**91.37**	**87.16**
Qwen-2.5-32B (Qwen et al., 2024)	zero-shot	51.11	54.91	50.63	34.20	27.21	41.39	43.24
	1-shot	70.34	72.22	64.31	57.91	44.05	67.38	62.70
	3-shot	73.03	75.38	67.82	63.18	48.10	70.63	66.36
	CoT	72.90	77.21	74.75	72.44	58.15	80.02	72.58
	SFT	**97.88**	**97.98**	**88.43**	**88.96**	**70.86**	**93.17**	**89.55**
Qwen-2.5-72B (Qwen et al., 2024)	zero-shot	48.57	51.64	46.54	37.16	27.45	37.76	41.52
	1-shot	68.93	70.34	62.32	57.95	44.23	66.36	61.69
	3-shot	72.34	73.67	65.56	**68.39**	**53.20**	72.98	67.69
	CoT	**77.52**	**82.36**	**77.80**	68.31	51.41	**79.64**	**72.84**
DeepSeek-V2-16B (DeepSeek, 2024)	zero-shot	29.44	33.90	27.92	18.57	11.46	12.80	22.35
	1-shot	64.35	66.42	52.76	47.04	28.04	49.20	51.30
	3-shot	**69.67**	**71.29**	**54.42**	54.03	**28.60**	**55.69**	**55.62**
	CoT	62.92	65.34	39.29	**61.41**	20.16	53.36	50.41
Yi-1.5-6B (01.AI et al., 2024)	zero-shot	32.41	35.70	28.36	18.75	10.25	7.46	22.16
	1-shot	45.40	48.28	38.09	31.31	19.41	22.61	34.18
	3-shot	49.88	53.82	43.24	36.37	21.73	32.25	39.55
	CoT	67.59	71.46	56.71	66.40	29.07	64.23	59.24
	SFT	**96.62**	**97.36**	**84.16**	**82.84**	**63.04**	**92.11**	**86.02**
Yi-1.5-34B (01.AI et al., 2024)	zero-shot	46.40	48.25	42.41	31.39	22.36	10.63	33.57
	1-shot	68.29	70.07	64.65	57.17	38.66	52.19	58.51
	3-shot	75.23	76.18	66.73	61.49	44.73	56.61	63.50
	CoT	74.16	79.01	69.62	68.67	48.24	72.78	68.75
	SFT	**97.82**	**97.95**	**87.60**	**89.22**	**70.80**	**91.45**	**89.14**

Table 6: F1 Score Overview for Error-Handling Tasks. Results are reported in percentage(%). "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr.", and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance.

Model	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
GPT-4o (OpenAI, 2023)	6.60	7.24	4.01	18.01	13.38	21.34	11.76
Gemini-1.5 (Google et al., 2024)	19.14	16.87	8.50	26.11	18.01	22.42	18.51
Claude-3.5 (Anthropic, 2024)	**29.15**	**25.60**	**16.87**	**32.57**	**24.02**	**42.47**	**28.45**
GLM-4 (GLM, 2024)	18.58	18.58	10.36	25.07	13.97	17.45	17.34
LLaMA-3.2-3B (Meta, 2024)	12.18	9.12	5.32	20.25	13.38	32.09	15.39
LLaMA-3.1-8B (Meta, 2024)	5.96	5.32	2.03	18.58	10.36	28.65	11.82
LLaMA-3.3-70B (Meta, 2024)	22.42	15.72	10.36	22.42	15.14	28.15	19.04
Qwen-2.5-7B (Qwen et al., 2024)	15.14	13.38	6.60	21.34	11.58	33.52	16.93
Qwen-2.5-32B (Qwen et al., 2024)	24.92	20.92	11.03	27.13	15.49	31.90	21.90
Qwen-2.5-72B (Qwen et al., 2024)	21.88	17.45	9.12	28.15	18.58	28.65	20.64
DeepSeek-V2-16B (DeepSeek, 2024)	8.50	6.60	3.36	18.58	12.18	15.14	10.73
Yi-1.5-6B (01.AI et al., 2024)	4.67	4.67	0.68	12.18	5.96	6.60	5.79
Yi-1.5-34B (01.AI et al., 2024)	16.30	14.56	10.36	24.55	15.14	7.87	14.80
Avg	15.80	13.54	7.58	22.69	14.40	24.33	16.39

Table 7: F1 scores for error-handling tasks in Language Error. Results are reported in percentage (%). "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr.", and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance.

Model	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
GPT-4o (OpenAI, 2023)	40.59	42.31	41.45	15.25	7.93	50.94	33.08
Gemini-1.5 (Google et al., 2024)	48.61	51.45	50.94	12.88	7.51	42.88	35.71
Claude-3.5 (Anthropic, 2024)	**72.51**	**76.84**	**75.96**	17.19	14.07	55.39	**51.99**
GLM-4 (GLM, 2024)	54.42	55.39	54.67	20.94	**16.42**	55.15	42.83
LLaMA-3.2-3B (Meta, 2024)	48.35	49.40	47.82	15.64	5.79	54.18	36.86
LLaMA-3.1-8B (Meta, 2024)	45.12	47.29	45.66	18.33	7.51	55.63	36.59
LLaMA-3.3-70B (Meta, 2024)	**53.20**	52.95	51.45	19.46	6.22	47.82	38.52
Qwen-2.5-7B (Qwen et al., 2024)	44.01	46.21	45.94	20.94	10.43	57.75	37.55
Qwen-2.5-32B (Qwen et al., 2024)	51.71	**53.72**	**53.72**	20.70	**12.55**	**70.16**	43.76
Qwen-2.5-72B (Qwen et al., 2024)	44.84	48.08	47.55	**23.12**	10.43	61.37	39.23
DeepSeek-V2-16B (DeepSeek, 2024)	28.35	26.98	25.25	14.07	4.48	21.68	20.14
Yi-1.5-6B (01.AI et al., 2024)	29.35	32.31	31.66	11.66	3.16	14.47	20.44
Yi-1.5-34B (01.AI et al., 2024)	39.11	38.52	37.62	13.68	7.08	21.68	26.28
Avg	46.17	47.80	46.90	17.22	8.74	46.85	35.61

Table 8: F1 scores for error-handling tasks in Incomplete Information. Results are reported in percentage (%). "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr.", and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance.

Model	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
GPT-4o (OpenAI, 2023)	73.13	74.65	71.84	48.31	41.81	22.60	55.39
Gemini-1.5 (Google et al., 2024)	77.34	**80.15**	70.25	43.39	31.18	11.41	52.29
Claude-3.5 (Anthropic, 2024)	77.10	77.10	**73.90**	55.96	52.93	27.95	60.82
GLM-4 (GLM, 2024)	68.89	69.44	67.23	58.56	49.04	26.04	56.53
LLaMA-3.2-3B (Meta, 2024)	69.17	66.95	61.98	44.17	34.74	17.97	49.16
LLaMA-3.1-8B (Meta, 2024)	67.79	67.51	62.88	49.40	39.39	17.44	50.74
LLaMA-3.3-70B (Meta, 2024)	**78.53**	77.58	73.13	**64.07**	**57.60**	**31.64**	**63.76**
Qwen-2.5-7B (Qwen et al., 2024)	68.07	68.62	63.48	59.51	43.00	27.00	54.95
Qwen-2.5-32B (Qwen et al., 2024)	74.49	74.49	69.20	61.92	47.67	29.20	59.50
Qwen-2.5-72B (Qwen et al., 2024)	73.64	73.64	69.44	61.67	50.48	27.95	59.47
DeepSeek-V2-16B (DeepSeek, 2024)	47.57	50.12	44.94	30.27	19.54	3.13	32.60
Yi-1.5-6B (01.AI et al., 2024)	51.89	52.93	44.17	34.30	15.29	0.63	33.20
Yi-1.5-34B (01.AI et al., 2024)	68.34	69.17	62.88	55.63	41.01	1.27	49.72
Avg	68.92	69.41	64.26	51.32	40.28	18.79	52.16

Table 9: F1 scores for error-handling tasks in Factual Errors. Results are reported in percentage (%). "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr.", and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance.

Model	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
GPT-4o (OpenAI, 2023)	42.73	56.47	44.73	45.85	31.89	18.68	40.06
Gemini-1.5 (Google et al., 2024)	67.80	**79.21**	**70.02**	48.85	37.64	9.80	52.22
Claude-3.5 (Anthropic, 2024)	**69.22**	76.77	67.39	52.25	38.26	**43.30**	57.87
GLM-4 (GLM, 2024)	61.36	69.22	55.50	55.25	37.02	19.83	49.70
LLaMA-3.2-3B (Meta, 2024)	49.12	52.50	36.08	44.44	25.36	29.54	39.51
LLaMA-3.1-8B (Meta, 2024)	47.77	55.99	40.97	48.58	28.51	28.51	41.72
LLaMA-3.3-70B (Meta, 2024)	68.82	75.87	64.23	**58.84**	**42.44**	37.33	**57.92**
Qwen-2.5-7B (Qwen et al., 2024)	49.65	60.00	45.57	52.50	30.22	28.17	44.35
Qwen-2.5-32B (Qwen et al., 2024)	56.31	68.64	55.57	54.33	37.45	24.95	49.54
Qwen-2.5-72B (Qwen et al., 2024)	53.77	62.48	49.65	55.25	35.45	25.00	46.93
DeepSeek-V2-16B (DeepSeek, 2024)	30.89	45.29	31.89	27.13	13.54	8.52	26.21
Yi-1.5-6B (01.AI et al., 2024)	41.56	48.04	28.86	37.02	17.13	5.47	29.68
Yi-1.5-34B (01.AI et al., 2024)	55.25	61.59	49.38	49.12	29.54	7.22	42.02
Avg	68.92	69.41	64.26	51.32	40.28	18.79	52.16

Table 10: F1 scores for error-handling tasks in Logical Errors. Results are reported in percentage (%). "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr.", and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance.

Secondary Error Category	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
Grammatical Errors	**13.59**	**11.76**	**9.90**	11.76	8.00	**41.32**	**16.06**
Punctuation Errors	3.77	5.61	1.90	15.93	12.61	9.17	8.17
Spelling Errors	2.13	4.21	0.00	**26.17**	**19.42**	10.20	10.36
Speaker or Person	43.94	46.27	46.27	11.01	9.26	**64.47**	**36.87**
TextContSubjSpec	26.28	28.78	25.00	**25.00**	**11.11**	56.63	28.80
Location	38.85	37.68	37.68	6.90	5.22	56.41	30.46
Date and Time Period	**53.24**	**56.34**	**56.34**	16.22	5.71	16.22	34.01
Relation Error	70.75	72.48	69.86	49.21	41.67	17.31	53.55
Entity Error	**77.71**	**79.78**	**77.01**	43.80	36.64	**30.16**	**57.52**
Circumstance Error	70.52	71.26	68.24	**51.66**	**46.58**	19.35	54.60
Fallacy of Relevance	42.55	54.90	42.55	37.96	29.23	8.62	35.97
Fallacy of Presumption	49.65	60.26	51.70	53.69	34.85	10.43	43.43
Fallacy of Defective Induction	**49.66**	**70.52**	**55.48**	**58.23**	**42.25**	**30.30**	**51.07**
Fallacy of Ambiguity	24.07	31.86	22.43	27.27	17.31	24.07	24.50

Table 11: F1 score of GPT-4o (OpenAI, 2023) in the secondary error categories. Results are reported in percentage (%). "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr.", and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance.

Model	Setting	Det.	Att. at Ident.	Acc. Ident.	Att. at Corr.	Acc. Corr.	Guid.	Avg
GPT-4o (OpenAI, 2023)	zero-shot	21.69	40.43	38.92	28.41	25.58	29.89	30.82
	1-shot	**83.87**	**83.53**	74.81	74.43	54.55	76.51	74.62
	3-shot	80.15	81.18	73.60	72.06	56.22	74.70	72.99
	CoT	80.00	81.91	**79.09**	**83.22**	**72.87**	**78.83**	**79.32**
Gemini-1.5 (Google et al., 2024)	zero-shot	29.38	54.37	51.52	19.88	13.58	18.52	31.21
	1-shot	57.48	67.63	64.41	40.98	30.43	33.65	49.10
	3-shot	59.11	72.00	70.04	38.83	30.43	39.20	51.60
	CoT	**72.02**	**74.67**	**73.40**	**73.35**	**73.65**	**73.94**	**73.51**
Claude-3.5 (Anthropic, 2024)	zero-shot	40.43	55.50	55.34	35.29	24.42	33.90	40.81
	1-shot	**86.49**	83.44	77.78	72.79	65.85	75.62	77.00
	3-shot	82.08	**85.80**	82.31	79.60	73.76	79.21	80.46
	CoT	81.59	82.15	**82.74**	**82.05**	**81.53**	**81.74**	**81.97**
GLM-4 (GLM, 2024)	zero-shot	35.36	55.77	51.26	35.68	25.58	30.34	39.00
	1-shot	**79.55**	**79.17**	68.31	77.74	66.39	**79.07**	75.04
	3-shot	74.47	76.45	70.59	78.00	70.72	78.69	74.82
	CoT	75.91	77.60	**75.81**	**78.62**	**73.26**	78.59	**76.63**
LLaMA-3.2-3B (Meta, 2024)	zero-shot	17.14	37.37	34.78	16.67	10.98	32.65	24.93
	1-shot	68.32	68.32	61.34	56.35	37.84	67.55	59.95
	3-shot	66.36	65.12	50.42	59.24	50.20	65.69	59.51
	CoT	72.60	70.99	63.56	76.73	60.50	73.31	69.62
	SFT	**97.28**	**96.95**	**80.95**	**95.97**	**72.88**	**91.70**	**89.29**
LLaMA-3.1-8B (Meta, 2024)	zero-shot	17.75	42.36	41.45	24.04	17.75	31.18	29.09
	1-shot	74.02	73.33	60.41	65.09	48.16	71.43	65.41
	3-shot	81.96	80.36	61.74	78.59	53.28	84.78	73.45
	CoT	69.91	71.16	73.29	74.33	69.18	74.39	72.04
	SFT	**91.58**	**91.24**	**81.27**	**90.58**	**75.52**	**85.60**	**85.97**
LLaMA-3.3-70B (Meta, 2024)	zero-shot	30.86	47.72	45.60	26.97	24.28	34.64	35.01
	1-shot	76.04	75.20	68.38	70.06	56.57	75.80	70.34
	3-shot	**88.75**	**87.93**	78.65	84.24	71.91	**88.52**	83.33
	CoT	85.81	83.28	**80.14**	**86.90**	**80.57**	86.18	**83.81**
Qwen-2.5-7B (Qwen et al., 2024)	zero-shot	18.60	39.20	39.36	20.43	13.33	30.77	26.95
	1-shot	41.03	60.34	51.71	41.75	31.69	54.71	46.87
	3-shot	68.00	68.97	57.01	60.50	42.05	62.40	59.82
	CoT	72.43	72.84	68.85	74.63	60.66	69.14	69.76
	SFT	**96.37**	**96.37**	**88.81**	**95.74**	**77.55**	**92.41**	**91.21**
Qwen-2.5-32B (Qwen et al., 2024)	zero-shot	20.00	39.80	38.50	17.98	15.76	31.35	27.23
	1-shot	57.01	64.78	59.29	60.58	50.73	66.40	59.80
	3-shot	68.27	77.26	66.67	72.09	56.62	67.70	68.10
	CoT	67.84	75.57	71.76	78.88	73.19	78.53	74.30
	SFT	**97.30**	**96.97**	**84.85**	**94.12**	**74.90**	**91.64**	**89.96**
Qwen-2.5-72B (Qwen et al., 2024)	zero-shot	28.57	47.00	44.56	20.93	18.07	33.33	32.08
	1-shot	49.77	61.86	56.34	51.79	46.31	62.50	54.76
	3-shot	**71.90**	75.70	66.96	68.31	60.44	73.52	69.47
	CoT	67.44	**76.47**	**72.50**	**71.38**	**66.67**	**76.81**	**71.88**
DeepSeek-V2-16B (DeepSeek, 2024)	zero-shot	13.84	36.26	31.03	14.37	8.86	12.74	19.52
	1-shot	46.08	56.25	46.70	34.87	20.24	40.82	40.83
	3-shot	60.00	**69.77**	**48.51**	55.75	29.38	**55.32**	53.12
	CoT	**63.93**	65.85	46.23	**68.94**	**43.65**	51.82	**56.74**
Yi-1.5-6B (01.AI et al., 2024)	zero-shot	19.63	25.00	25.00	14.63	11.39	4.00	16.61
	1-shot	34.25	43.43	37.84	22.22	16.87	22.49	29.52
	3-shot	32.22	45.45	38.04	28.09	16.25	27.91	31.33
	CoT	65.91	65.32	52.02	71.35	45.95	66.67	61.20
	SFT	**96.99**	**96.35**	**84.94**	**94.77**	**77.91**	**90.78**	**90.29**
Yi-1.5-34B (01.AI et al., 2024)	zero-shot	28.07	43.62	38.46	21.30	15.85	7.84	25.86
	1-shot	57.58	64.29	60.87	42.62	34.83	39.65	49.97
	3-shot	69.17	75.59	60.91	55.96	38.95	54.29	59.15
	CoT	68.60	73.73	60.71	75.09	55.75	70.20	67.35
	SFT	**98.99**	**98.66**	**89.55**	**97.35**	**73.42**	**95.44**	**92.24**

Table 12: F1 scores for error-handling tasks in human evaluation. Results are reported in percentage (%). "Det.", "Att. at Ident.", "Acc. Ident.", "Att. at Corr.", "Acc. Corr.", and "Guid." stand for error detection, attempt at error identification, accurate identification, attempt at error correction, accurate error correction, and error guidance.