Language Models can Categorize System Inputs for Performance Analysis

Anonymous ACL submission

Abstract

Language model systems are used to process diverse categories of input requests, ranging from improving creative writing to solving programming challenges. It would be useful to know which categories they are good at. However, existing evaluations compare model performance on pre-defined categories, failing to reflect a system's performance on finer-grained or novel ones. We propose to automatically search for finer-grained categories based on inputs where a system performs well or poorly, and describe them in natural language. To search for these categories, we propose a large number of candidate category descriptions, e.g. "Communication Improvement", find the subset of inputs that match the category descriptions, and calculate the performance on these categories; then we sort these categories based on their performance, thereby highlighting those that score high or low. As one application, we apply our method to compare LLaMA 3-70B and Claude 3 Opus, which have similar Elo-ratings on Chatbot Arena; our method finds the former is weaker at making text more professional and humorous while better at providing psychological insights, depicting a more nuanced picture of model performance. [1]

1 Introduction

Language models (LMs) can perform a wide range of tasks, from writing to programming and solving math problems. If a user wants to make their text more professional, they will prefer an LM that is good at writing; its coding ability irrelevant.

However, the current evaluation paradigm falls short in offering detailed insights, as it only provides a single numerical score for broad categories. For example, ChatBot Arena (Chiang et al., 2024) uses Elo ratings based on pairwise comparisons to rank models, with Gemini-1.5 Pro, LLaMA 3-70B, and Claude 3 Opus (Reid et al., 2024; Meta, 2024; Anthropic, 2023) having similar ratings. Are their performances uniformly similar in programming and improving writing?

Indeed we can create a pre-defined category called "communication improvement" beforehand, but it is difficult to anticipate categories in practice, especially as new use cases constantly emerge. Fixed datasets such as GSM8K, BigCodeBench or DS-1000 (Cobbe et al., 2021; Lai et al., 2022; Zhuo et al., 2024) can provide a more detailed picture of model abilities; however, they often focus on high-level tasks such as grade-school math or coding questions, without revealing performances in finer categories, e.g. the proficiency in computer networking. A solution could be to manually propose new categories by skimming through thousands of samples, but this is both time-consuming and labor-intensive. We would like an automatic and adaptive categorization procedure.

To address these requirements, we constructed the Automated System Input Categorization Scheme (ASICS), a method that adaptively explains input categories in natural language and assigns scores to each, with the highest and lowest scores highlighted. ASICS takes in system inputs and the LM's performance on each system input (e.g whether the response to the input is correct). It outputs a list of categories, each described by a natural language string; each category is associated with the LM's average performance, with the best and worst categories highlighted. ASICS operates in three stages: propose, falsify, and score, as illustrated in Figure 2. First, given a set of system inputs with both correct and incorrect answers, the *category proposer* generates natural language descriptions of the categories that appear across them. Second, the *falsifier* ensures that each category is unambiguous and understandable; it does this by evaluating whether an LM can confidently determine which inputs match the generated categories. Finally, the *category scorer* evaluates each category

[1] We provide code in an anonymous GitHub repository.

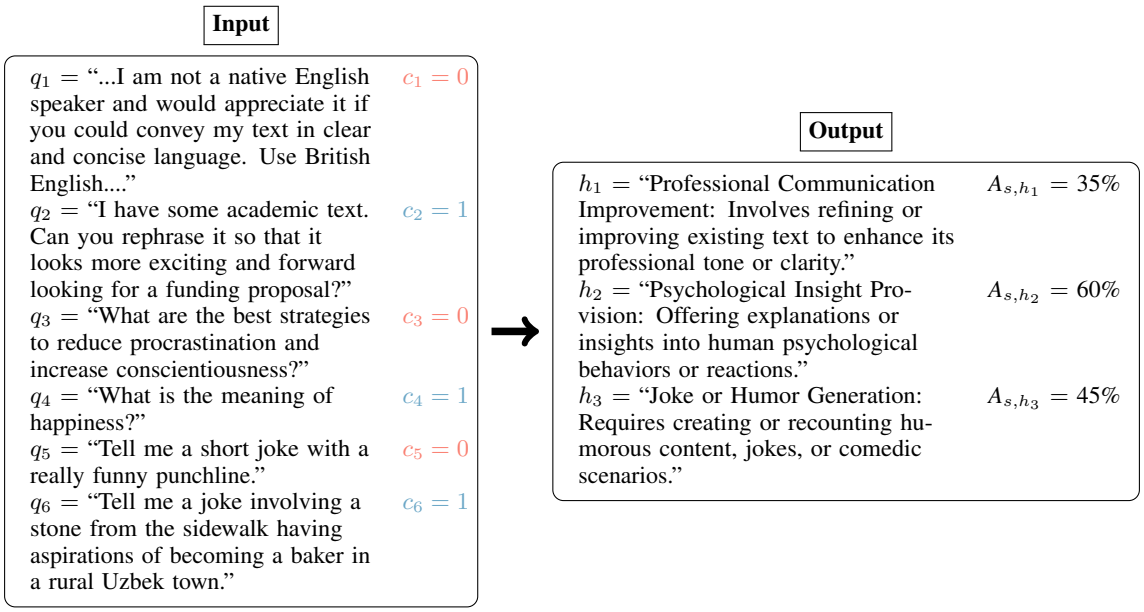

Figure 1: Categories generated by ASICS, based on the human preference dataset (HPD). The blue represents questions answered correctly and the red represents questions answered incorrectly. The score represents the percentage of times that LLaMA 3-70B is preferred over Claude 3 Opus for all prompts that represent the category of "Professional Communication Improvement" and "Psychological Insight Provision". The baseline score is 52%, indicating that Claude 3 Opus performs better than LLaMA 3-70B at "Professional Communication Improvement".

by finding the inputs that match the category and computing their average performance.

We applied ASICS to GSM8K, BigCodeBench and a human preference dataset (HPD)[2] that compares different LMs. On HPD, ASICS discovers categories where humans prefer LLaMA 3-70B, and Claude 3 Opus. For example, Claude 3 is preferred over LLaMA for "Professional Communication Improvement" while LLaMA is preferred for providing "Psychological Insights" (Section 5.1). This demonstrates that ASICS can generate novel categories. On BigCodeBench, ASICS discovers categories where LLaMA 3 is better than GPT-4 despite having a 10% lower overall score, such as, 'Network Operations' and 'Web Scraping' (Section 5.3). This demonstrates that ASICS can generate fine-grained categories, providing more detailed insights into system abilities than datasets alone. Across all datasets, ASICS identifies both categories that can be validated by prior works, as well as new categories that may motivate future investigation.

We can already see interesting results with only thousands of data points, and scaling this up to larger datasets could yield more interesting insights, thus helping humans have a global understanding of LMs' performance. Finally, we explore future applications of ASICS in other fields like education (Section 6).

[2]The data was provided to us by Chiang et al. (2024).

2 Related Work

Language Models and Inductive Reasoning ASICS generates categories by recognizing patterns within system inputs. Our approach builds on work that uses language models (Radford et al., 2019; Anthropic, 2023; OpenAI, 2023) to generate hypotheses from individual instances (Wang et al., 2024; Qiu et al., 2024; Steinhardt, 2023). Recent work indicates that under certain conditions, language models can perform inductive reasoning, identifying patterns in a collection of text data points and articulating them using language (Honovich et al., 2022). Zhong et al. (2022) and Singh et al. (2023) show that LMs are able to discover patterns in datasets. Tong et al. (2023) generate categories by identifying patterns in inputs that produce the same output, but should not in multimodal models. Zhong et al. (2023) automatically discover differences between two large corpora using language models that are relevant to a downstream goal. Wang et al. (2023) use language models to generate clusters describing an input corpus according to a user defined goal. Our work differs by demonstrating the ability of language models to categorize patterns in human preference data.

Analyzing Language Model Weak Points Benchmarks related to math, coding, and reasoning reveal inputs that the model errs on, but does not find common patterns among these inputs (Hendrycks et al., 2021; Kasai et al., 2023; Lai et al., 2022; Cobbe et al., 2021). Jones and Steinhardt (2022) employ cognitive biases to detect and evaluate systematic flaws in code models. McCoy et al. (2023) study language model failures in the context of the task they were trained on, next word prediction, and find that LM accuracy is significantly influenced by the probability of task, target output, and input. Other work uses humans to hypothesize potential failure modes such as biases (Maluleke et al., 2022), stereotypes (Blodgett et al., 2021), and then test for their existence in generative models. Our work is unique as it is the first to automatically generate and score categories—some substantially worse than baseline performance—across various natural language datasets without prior knowledge of the tasks.

Automated Discovery Much work has been done to automatically discover patterns in empirical data. Linear regression analyzes the effect of each real-valued feature by interpreting the learned weights (Gelman et al., 2020). Topic models (Sobhani et al., 2024; Sridhar et al., 2022) uncover latent topics in text corpora. Causal forests learn to partition data into relevant subgroups to estimate heterogeneous treatment effects (Wager and Athey, 2018). Ludwig and Mullainathan (2024) use machine learning for hypothesis generation and develop a procedure to produce hypotheses about what features of defendants mug shot influence judicial decisions. Li et al. (2024) use large language models to automate the discovery of statistical models. Our work differs by categorizing natural language data into some novel categories, as well as existing ones, and scoring the model performance across these categories.

Evaluating Language Models Frameworks such as HELM, BigBench, and ChatbotArena conduct comprehensive evaluations of language models. HELM offers a broad evaluation across their capabilities, limitations, and risks across diverse scenarios and metrics (Liang et al., 2023). BigBench emphasizes task diversity and the ability of models to generalize across a wide range of challenges (Srivastava et al., 2023). Chatbot Arena is an open platform for evaluating LMs based on human preferences using pairwise comparisons and crowdsourcing (Chiang et al., 2024). HELM and BigBench cover broad categories like physics or childhood development but fail to capture finer details within them, while Chatbot Arena only provides comparisons across high-level categories such as different languages and prompt lengths without identifying fine-grained categories within the prompts. Our approach builds on existing methods by using ASICS to automatically and adaptively generate fine-grained, natural language descriptions from datasets and human preference data, identifying novel categories beyond traditional benchmarks.

3 The ASICS Pipeline

We describe ASICS, a method to automatically identify fine-grained categories from system inputs with correct and incorrect answers. We use s to denote the target language model system to be evaluated and $\mathcal{Q}$ the set of inputs where s was evaluated. We define $c_{q,s} = 1$ if the system s answers the input q correctly, and 0 otherwise. In the context of the HPD $c_{q,s}$ represents when s is preferred. ASICS takes in $\mathcal{Q}$ and c, and outputs a set of category descriptions $\mathcal{H}$, where each category is associated with a performance score. The highest and lowest scoring categories will be highlighted to indicate where the system performs exceptionally well or poorly.

ASICS relies on three components: the *category proposer*, the *falsifier*, and the *category scorer*. The *category proposer* suggests categories $\{h_1, \ldots, h_k\}$ based on $\mathcal{Q}$. The *falsifier* ensures that each category h_i is comprehensible by evaluating whether a language model can confidently align inputs with the generated categories. An input is considered aligned if it possesses the features described in the category (e.g., an input aligns with the "Conceptual Explanation in Mathematics or Physics" category if it contains features related to math or physics, as shown in Table 1). The *category scorer* assesses the alignment between all inputs and categories, and scores the performance of s on the questions aligned with each category. Experimental details for this section, including the prompts, can be found in Appendix A.

3.1 Category Proposer

Based on $\mathcal{Q}$ that s was evaluated on, ASICS proposes fine-grained categories. To do so, ASICS prompts a LM with the prompt shown in Figure 3 (top) along with batches of $\mathcal{Q}$. The model then

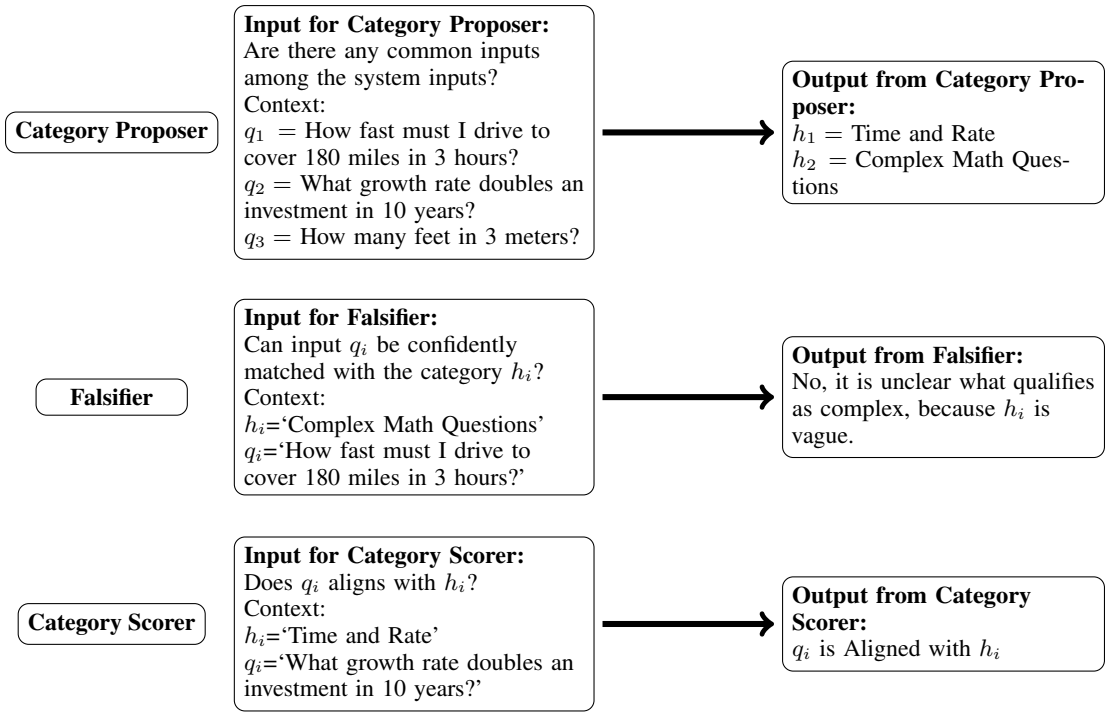

Figure 2: The prompt template for all components in our system. The ASICS pipeline begins by taking as input the set of questions a system was evaluated on. Each row represents a stage in the process: (1) generating categories based on the questions (Category Proposer), (2) evaluating if the categories can be confidently matched to the questions (Falsifier), and (3) determining if the questions align with the category (Category Scorer).

generates categories $h_1, \ldots, h_k$ for each batch of $\mathcal{Q}$, which we automatically parse and add to the set of categories $\mathcal{H}$.

For example, if we prompt GPT-4 with inputs where LLaMA-3 is less preferred such that $q_i \in \mathcal{Q} \mid c_{q_i,s} \neq \mathbb{1}$, one of the categories h_i generated by the proposer is "Conceptual Explanation in Mathematics or Physics." The number of categories generated for each batch depends on the number of inputs in that batch, and the size of $\mathcal{H}$ depends on the size of inputs $\mathcal{Q}$. For all experiments we use GPT-4, Claude 3 Opus or Claude 3.5 Sonnet as the category proposer.

3.2 Category Scorer

ASICS then scores the performance of s, on each category. To do this, ASICS prompts a LM to determine whether each input in $\mathcal{Q}$ aligns with each category h_i. We pass the prompt in Figure 3 (mid) to a LM to evaluate the input-category alignment.

We then compare these scores to s's baseline performance on all inputs, denoted as $A_{s,Q}$. We define A_{s,h_i} to be the accuracy of s on category h_i. If h_i is "Conceptual Explanation in Mathematics or Physics" and s is LLaMA-3, then A_{s,h_i} represents how often is LLaMA-3 preferred on math or physics questions. We use Claude 3 Opus or Claude 3.5 Sonnet as the *category scorer*.

Aligned Inputs
Prove that if $AB - BA = A$, then $\det(A) = 0$
Can you provide a Taylor's series for $e^x \sin(x)$?
If the gravity of a black hole is so strong that it stops time, how can it spin?
Explain simply the Landauer principle

Table 1: Inputs that were classified as aligned with the category "Conceptual Explanation in Mathematics or Physics." In Section 4 we show with human annotators that the inputs Claude 3 Opus classifies as aligned with each category are in-line human opinions.

3.3 Refine

It is important that $\mathcal{H}$ is refined to ensure that each category is distinctive, unambiguous, and deviates substantially from $A_{s,Q}$.

Deduplication After generating categories with the *category proposer* we embed the title of each category with text-embedding-ada-002 (OpenAI, 2022) and calculate cosine similarity scores. Any category h_i with a cosine similarity score above τ with another category h_j is removed.

Falsifier The *category scorer* requires each category to be unambiguous to accurately determine

Category Proposer Prompt

Here are system inputs for you to memorize.
{n individual system inputs}
Using these specific examples, are there any common features among the system inputs? [...]

Category Scorer Prompt

Determine if the question is asking for information described in the task [...].
Question: {}
Task Title: {}
Task Description: {}

Falsifier Prompt

Assess the alignment between the provided question and task. [...] Assess if the task description is too vague, ambiguous, or broadly interpreted to make a confident judgment on alignment [...].
Question: {}
Task Title: {}
Task Description: {}

Figure 3: (Top) Prompt used by ASICS to propose categories. In the prompt text, [...] indicates further text that is omitted here for space. (Mid) Prompt used by ASICS to evaluate if a question belongs to category. (Bottom) Prompt used by ASICS to falsify categories.

which inputs are aligned with it. However, ASICS can generate categories that are unclear. For example, on the HPD, the *category proposer* generated: "Synthesizing Knowledge from Hypothetical Constructs or Abstract Scenarios." It is unclear what qualifies as an "abstract scenario," making it difficult to determine if an input is aligned with this category. To ensure that each discovered category can be confidently scored, we assess whether, given a random sample from $\mathcal{Q}$, a LM can determine if an input aligns with a category. For a subset of 20 inputs from $\mathcal{Q}$ and a given category h_i, ASICS queries a LM with the prompt in Figure 3 (bottom).

Should the LM be incapable of confidently verifying two or more inputs with h_i, we remove h_i from $\mathcal{H}$. For all experiments we use Claude 3 Opus or Claude 3.5 Sonnet as the *falsifier*.

Salient Categories We also refine $\mathcal{H}$ after the *category scorer* step by ignoring categories with scores that have all small deviation from the overall score.

4 Human Evaluation of ASICS

In this paper, ASICS uses Claude 3 Opus as the *category scorer*. To ensure that Claude 3 Opus accurately identifies which questions belong to each category, we measure its agreement with human annotators on GSM8K and the HPD.

Annotators are presented with a category and two corresponding questions: one randomly sampled from a dataset of questions classified by Claude as aligned with the category, and the other randomly sampled from a dataset of questions classified as unaligned. Each participant evaluates which of the two questions best matches the given category. A success for the *category scorer* is when the annotator and Claude 3 Opus select the same question as being aligned with the category. In the second human evaluation experiment, we assess whether a randomly sampled input question from a dataset—balanced with an equal number of questions Claude classified as aligned and unaligned for a specific category—is aligned with that category. We experiment with two different evaluation mechanisms involving crowd-workers and experts.

Evaluation via Crowdsourcing and Experts. We conducted the human evaluation using the Amazon Mechanical Turk platform and expert volunteers. For the HPD, we focused on questions passed to LLaMA 3-70B and Gemini-1.5 Pro, pairing questions classified as aligned with each of the 8 best-scoring (those with scores furthest away from the baseline performance) categories with unaligned questions. Using majority voting to determine the ground truth on 321 prompt pairs, we found that Claude-3 Opus achieved a score of 92%. For the GSM8K data, we evaluated 100 pairs of questions for 3 randomly sampled categories, finding that Claude 3 Opus achieved a score of 85%.

In addition, we recruited three expert volunteers from an engineering graduate school to evaluate 9 categories, each with 50 pairs of questions. Using majority voting to determine the ground truth, Claude 3 Opus achieved a score of 92%, with an inter-annotator agreement of 85%.

On the second human validation experiment, using majority to determine the ground truth on 200 prompt pairs from the HPD, we found that the annotator and Claude 3 Opus achieve an agreement rate of 83% over 200 (sample, category) pairs. More details of experiments and full descriptions of the categories can be found in Appendix E.

5 Results

We apply ASICS on several datasets:

1. In Section 5.1, we apply ASICS to the HPD and discover categories where humans prefer

responses from LLaMA 3-70B, and Claude 3 Opus.

2. In Section 5.2, we apply ASICS to GSM8K and discover categories where different Claude models, despite having similar overall performance on GSM8K, perform differently.
3. In Section 5.3, we apply ASICS to BigCodeBench and discover categories where CodeLlama-70B outperforms GPT-4o, even though the former has a lower overall score.

5.1 Finding Categories in Human Preferences

The HPD dataset includes inputs that are passed to LLaMA 3-70B and Claude 3 Opus, with human labels indicating the preferred output. We chose to compare this specific pairs because they have a large number of preference labels (e.g., 770) in the HPD. We use Claude 3.5 Sonnet for each component of the ASICS pipeline.

We aim to find categories with scores that deviate significantly from baseline performance. To establish baseline scores, we calculated the percentage of times LLaMA 3's output was preferred over Claude. In HPD, LLaMA 3 was preferred 52% of the time over Claude.

We split 30% of the dataset into a category selection set and the remaining 70% into an evaluation set. Categories are generated based on the category selection set, which is also used for falsification and scoring. We focus on categories that meet the following criteria: the accuracy difference from the baseline exceeds 10%, there are more than 16 samples per category, less than 10% of samples are classified as uncertain, and redundant categories with $\tau = 0.82$ are removed. After applying these criteria we then evaluate using the samples in the evaluation set. In Table 3 we present the categories and their scores that ASICS generates on the HPD. We find that Claude performs best on social tasks such as "Professional Communication Improvement" and "Generating Pseudoprofound Content," while LLaMA performs better on "Psychological Insight Provision", although it is not statistically significant.

We conduct significance testing to evaluate whether the difference between the category score and overall score is statistically significant. Multiple hypothesis testing with Bonferroni correction is conducted to test for significance. While some differences are not statistically significant, the limited size of the evaluation set (537 samples) suggests that running ASICS on larger datasets could yield more significant results. Despite some differences not being statistically significant, we believe the categories offer valuable insights for future exploration.

Positive	meaning, ten, side, take, future, line, states
Negative	joke, episode, dog, series, city, generate, apple

Table 2: The terms most predictive of human preference labels based on regression analysis show no meaningful patterns among the terms with the largest coefficients.

Unigram and Clustering Analysis To the best of our knowledge, no other work compares model performance across the fine-grained categories ASICS discovered. We conduct an analysis to evaluate whether the categories discovered by ASICS could be discovered by predicting the human preference label based on the inputs. We display the unigrams in Table 2 with the most positive and negative coefficient values. We are unable to discern any patterns among these unigrams. We also perform a clustering analysis with k-means based on the sentence-BERT embeddings of each input. We display the inputs in the first cluster in Table 4. We are unable to discern any patterns using the clustering method. Thus, we believe many of these categories are novel and can not be described by alternative methods. Additional experimental details can be found in Appendix C.

Ablations We rely on the *category proposer* to generate categories, while the *falsifier* and *category scorer* are also crucial components of ASICS. From 1,000 samples of the HPD, the proposer generates 128 categories. After applying the *falsifier* with a 10% uncertainty threshold, we are left with 67 categories. For example, the category "Multi-Step Problem Solving with Domain Knowledge" was filtered out due to 28% uncertainty (73 out of 265 samples).

The *category scorer* step allows us to consider categories that deviate substantially from the mean (e.g., either 15% more positive or negative) and thus are more insightful for practitioners. This, along with deduplication, trims the 67 categories after falsification to 7 categories

5.2 Evaluating Claude on Math Categories

GSM8K is a dataset covering a wide range of grade school math questions. We test Claude 1.2, 2.0, 2.1, and Haiku on its test set. We generate categories

Category	Score
Joke or Humor Generation	45.00% $^{-7.45}$
Professional Communication Improvement	35.42% $^{-17.91*}$
Psychological Insight Provision	60.42% $^{+8.99}$
Fictional Universe Knowledge Retrieval	40.63% $^{-11.79}$
Logical Reasoning and Deduction	46.88% $^{-5.57}$
Overall Score	52.45%

Table 3: Categories ASICS finds based on the HPD. The values in the upper right corner of each row in the score column show deviation from baseline performance, and asterisk (*) indicates statistical significance after controlling for multiple hypothesis testing.

What is an entanglement witness?
What is the connection between capybaras and meth?
Who is Garrus Vakarian and what is he known for?
What is a definitive test to diagnose amblyopia?

Table 4: Top questions in the first k-means cluster ($k = 20$) via SBERT show no identifiable patterns.

based on questions that Claude 1.2 answered incorrectly and use GPT-4 as the *category proposer*.

We present the results for the Claude models when tested with zero-shot CoT prompting (Kojima et al., 2022) in Table 5. We find that all models perform similarly on GSM8K, motivating us to assess how their performance varies across the categories generated by ASICS in Table 5.

Due to the small number of samples in the GSM8K test set that align with each category (e.g., fewer than 100), even a slight difference in the number of correctly answered questions for each model could significantly impact the performance metrics. Therefore, we generate additional samples to improve the confidence of our model comparisons. We create 370 new synthetic questions using Claude 3 Opus, focusing on the four categories identified by ASICS.

We find that model performance drops substantially when tested on questions within the categories identified by ASICS, indicating these are weak points for the models. We find that on fine-grained categories such as "Temporal and Age-Related Calculation" performance drops by over 30%, and for "Predicting Future Values Based on Growth or Change Rates" performance drops by over 22% for all models. ASICS automatically identifies these failures and also uncovers new systematic failures, not previously reported, such as "Conversion and Unit Management."

Despite similar overall performance, models exhibit varying performance across different categories (Table 5). For example, while Claude 3 Haiku has the highest overall performance, it performs 5% worse than Claude 2.1 on the category "Predicting Future Values Based on Growth or Change Rates," with its performance dropping 7% more than Claude 2.1. Claude 2.1, while third in overall performance, outperforms others on categories such as "Elapsed Time Calculation" by multiple percentage points. This underscores the limitations of selecting a model based solely on a single numerical summary for the entire dataset and highlights the importance of fine-grained evaluations for tasks with pre-existing datasets. Full descriptions of each category and additional experimental details can be found in Appendix B.

5.3 Finding Categories in BigCodeBench

BigCodeBench is a code generation benchmark encompassing diverse problems from cryptography to data visualization. We evaluate CodeLlama-70b, gpt-4o, and deepseek-coder-33b as we wanted a mix of open and closed models from different model families. We use the results reported by Zhuo et al. (2024). We use Claude 3.5 Sonnet for each component of the ASICS pipeline.

In Table 6 we present the categories ASICS generates and their scores. For each model and after adjusting for multiple hypothesis testing with bonferrini correction, we find categories whose score deviates significantly from the overall model score. On questions related to 'Network Operations' and 'Web Scraping and API Interaction', the performance of GPT-4o drops by over 20%, and Deepseek also experiences a similar performance drop on these categories. Despite performing 11% worse than GPT-4o overall, Code-Llama outperforms, GPT-4o on 'Network Operations' and 'Web Scraping' declines. This highlights the limitations of relying on a single numerical summary for broad benchmarks such as BigCodeBench and the importance of getting a more fine-grained understanding of LM performance. As the use of language models for coding tasks continues to grow, ASICS can provide valuable insights into specific and semantically coherent categories that require improvement.

Unigram Analysis We conduct an analysis to evaluate if the categories discovered by ASICS could be discovered by predicting whether the question was answered correctly based on the inputs. We display the unigrams in Table 7 with

Category	Claude 1.2	Claude 2.0	Claude 2.1	Claude 3 Haiku
Conversion and Unit Management	67.66% $^{-19.88}$	74.13% $^{-15.10}$	74.39% $^{-13.68}$	**77.02%** $^{-13.12}$
Temporal and Age-Related Calculations	56.61% $^{-30.93}$	50.93% $^{-38.30}$	55.79% $^{-32.28}$	**59.83%** $^{-30.31}$
Predicting Future Values Based on Growth or Change Rates	53.37% $^{-34.17}$	63.32% $^{-25.91}$	**65.50%** $^{-22.57}$	60.65% $^{-29.49}$
Elapsed Time Calculation	68.18% $^{-19.36}$	74.13% $^{-15.10}$	**77.63%** $^{-10.44}$	74.65% $^{-15.49}$
GSM8K Baseline	87.54%	89.23%	88.07%	**90.14%**

Table 5: Performance across Claude models for different GSM8K categories. The values in the upper right corner of each row in the score column represent the deviation from baseline.

Category	GPT-4o	CodeLlama-70B	Deepseek-coder-33B
Combination and Permutation Generation	**69.77%** $^{+9.89}$	**69.77%** $^{+20.72*}$	62.79% $^{+12.23}$
Network Operations	37.29% $^{-23.19*}$	**38.98%** $^{-10.64}$	27.12% $^{-23.46*}$
Web Scraping and API Interaction	39.58% $^{-20.92*}$	**41.67%** $^{-7.37}$	33.33% $^{-17.24*}$
Overall Score	**60.48%**	49.05%	50.57%

Table 6: Performance for different models and various categories. Values in the upper right corner of each cell show deviation from the overall score for that model. Categories with an asterisk (*) indicate statistical significance.

Positive	categories, data_matrix, delay, ctypes, range, arr, name, input_str, choice, unicodedata
Negative	array_size, target_value, zipfile, random_state, db_path, backup_dir, my_list, urllib, subprocess, sys

Table 7: The most predictive features for GPT-4o's correct answers from regression analysis are insightful but less meaningful than ASICS-generated categories.

the most positive and negative coefficient values. The features are informative about libraries that models struggle with, for example'urllib' indicate that the GPT-4o struggles with network requests. However, unigram analyses are less actionable and semantically-coherent compared to the categories ASICS generates. It is also unclear how complex categories like 'Combination and Permutation Generation' could be discovered via this analysis.

These results highlight the usefulness of more fine-grained analyses of systems. As the use of language models for coding tasks continues to grow, ASICS can provide valuable insights into specific and semantically coherent categories that require further improvement.

6 Discussion

We developed ASICS, a method that automatically finds categories in inputs and describes them in natural language. To generate more novel categories ASICS can be deployed at industry scale where (prompt, evaluation) pairs are easier to obtain. In these cases, the users write the prompt themselves, and the LM platform can (implicitly) ask the users to evaluate the responses. For example: ChatBotArena is an open platform that mimics chatbot deployment. Within 15 months, it has collected 1.5M (prompt, evaluation) pairs. ChatGPT allows users to provide feedback if an LM's response is bad, and also allows users to label whether an entire conversation is helpful or not. This will likely gather many more such pairs than ChatbotArena. character.ai receives ∼1.7B queries per day, and a proxy evaluation is whether the user continues the conversation.

In principle, ASICS can identify patterns in instance-level data points if the data can be described using natural language. As models are increasingly deployed in domains where human evaluation is time-consuming and challenging, adaptive and automatic methods like ASICS can provide insights into potential model failures in these settings. As language models continue to improve and scale, ASICS's performance is likely to improve as well.

Broader Applications Real-world systems could also be studied with ASICS. ASICS could be used to find patterns among successful and unsuccessful Y Combinator applicants (Bhalotia, 2022). Teaching at the Right Level, which aims to determine the learning level students are at and then teach them at their level, has proven to be an effective way to improve learning outcomes (Banerjee et al., 2007). Teachers could use ASICS to find categories among questions students answer incorrectly.

Limitations

ASICS is not guaranteed to find all categories within the data it analyzes. In all experiments, we used Claude 3 Opus as the category scorer, and we do not know how alignment with human annotators might differ for other models. The humans in the HPD are not randomly sampled, so we cannot claim these results will hold for a different sample representing the same task (Freedman, 2009) Claude 3 Opus or GPT-4 is used as the category proposer, the categories discovered with other models could differ. The generated categories are a function of the LM used to identify them, thus any biases from the LM can effect the categories generated. Language models often generate outputs stochastically, which means that categories may vary across different runs. Additionally, the datasets we evaluate are predominantly in English, so the categories may vary across different languages, and the ability of the pipeline to discover relevant categories may decrease. Using models that rely on API's can be time-consuming, depending on the dataset being evaluated. The categories we identify are based on correlations within the data, they do not indicate causation regarding the reasons behind these performance differences.

Ethical Statements

We acknowledge that ASICS could be applied to study real-world systems or other consequential data that can be described in natural language. We emphasize the importance of human judgement when applying ASICS. It is also important not to misinterpret correlation as causation in categories ASICS finds, as this could reinforce societal biases.

References

Anthropic. 2023. Introducing claude. Anthropic.

Abhijit V. Banerjee, Shawn Cole, Esther Duflo, and Leigh Linden. 2007. Remedying education: Evidence from two randomized experiments in india. *The Quarterly Journal of Economics*, 122(3):1235–1264.

Akshay Bhalotia. 2022. Yc company scraper.

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. Stereotyping norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1004–1015.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv*.

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

David A. Freedman. 2009. *Statistical Models and Causal Inference: A Dialogue with the Social Sciences*. Cambridge University Press, Cambridge.

Andrew Gelman, Jennifer Hill, and Aki Vehtari. 2020. *Regression and Other Stories*. Analytical Methods for Social Research. Cambridge University Press, Cambridge.

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

Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. 2022. Instruction induction: From few examples to natural language task descriptions. *arXiv preprint arXiv:2205.10782*.

Erik Jones and Jacob Steinhardt. 2022. Capturing failures of large language models via human cognitive biases. *Advances in Neural Information Processing Systems*, 35:11785–11799.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. Realtime qa: What's the answer right now? In *Advances in Neural Information Processing Systems 36*, pages 49025–49043.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Scott Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2022. Ds-1000: A natural and reliable benchmark for data science code generation. *Preprint*, arXiv:2211.11501.

Michael Y. Li, Emily B. Fox, and Noah D. Goodman. 2024. Automated statistical model discovery with language models. *arXiv*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, et al. 2023. Holistic evaluation of language models. *arXiv*.

Jens Ludwig and Sendhil Mullainathan. 2024. Machine learning as a tool for hypothesis generation. *The Quarterly Journal of Economics*, 139:751–827.

Vongani H Maluleke, Neerja Thakkar, Tim Brooks, Ethan Weber, Trevor Darrell, Alexei A Efros, Angjoo Kanazawa, and Devin Guillory. 2022. Studying bias in gans through the lens of race. In *Computer Vision–ECCV 2022: 17th European Conference*, pages 344–360, Tel Aviv, Israel. Springer.

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L. Griffiths. 2023. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv.org*.

Meta. 2024. Meta llama 3. `https://ai.meta.com/blog/meta-llama-3/`. Accessed: 2024-06-12.

OpenAI. 2022. text-embedding-ada-002. `https://platform.openai.com/docs/guides/embeddings`.

OpenAI. 2023. Gpt-4 technical report. Technical report.

L. Qiu, L. Jiang, X. Lu, M. Sclar, V. Pyatkin, C. Bhagavatula, B. Wang, Y. Kim, Y. Choi, N. Dziri, and X. Ren. 2024. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. In *The Twelfth International Conference on Learning Representations*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv*.

Chandan Singh, John X. Morris, Jyoti Aneja, Alexander M. Rush, and Jianfeng Gao. 2023. Explaining patterns in data with language models via interpretable autoprompting. *arXiv*.

Dominic Sobhani, Amir Feder, and David Blei. 2024. Environment-adjusted topic models. In *ICLR 2024 Workshop on Data-centric Machine Learning Research (DMLR): Harnessing Momentum for Science*.

Dhanya Sridhar, Hal Daumé, and David Blei. 2022. Heterogeneous supervised topic models. *Transactions of the Association for Computational Linguistics*, 10:732–745.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv*.

Jacob Steinhardt. 2023. Language models as statisticians, and as adapted organisms. Simons Institute.

Shengbang Tong, Erik Jones, and Jacob Steinhardt. 2023. Mass-producing failures of multimodal systems with language models. *Advances in Neural Information Processing Systems*, 36:29292–29322.

Stefan Wager and Susan Athey. 2018. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242.

R. Wang, E. Zelikman, G. Poesia, Y. Pu, N. Haber, and N. D. Goodman. 2024. Hypothesis search: Inductive reasoning with language models. In *The Twelfth International Conference on Learning Representations*.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. Goal-driven explainable clustering via language descriptions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10626–10649, Singapore. Association for Computational Linguistics.

Ruiqi Zhong, Charlie Snell, Dan Klein, and Jacob Steinhardt. 2022. Describing differences between text distributions with natural language. In *Proceedings of the 39th International Conference on Machine Learning*, pages 27099–27116. PMLR.

Ruiqi Zhong, Peter Zhang, Steve Li, Jinwoo Ahn, Dan Klein, and Jacob Steinhardt. 2023. Goal driven discovery of distributional differences via language descriptions. *Advances in Neural Information Processing Systems*, 36:40204–40237.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, et al. 2024. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. *ArXiv*, abs/2406.15877.

A Prompts Used in ASICS

In this section, we provide the prompts used in ASICSfor the category proposer, falsifier, and category scorer.

A.1 Prompt for Category Scorer

To propose categories we ask Claude 3 Opus or GPT-4 to find common categories in system inputs.

Prompt

I will provide a series of data for you to remember. Subsequently, I will ask you some questions to test your performance! Here are questions for you to memorize.
{n system inputs}
The above questions were answered by a system.Using these specific examples, are there any tasks that are shared among the questions the system answers?
For each identified task, please offer a detailed and precise description. Highlight the particular characteristics of the tasks in the questions the system answered.
Please focus on achieving a high level of specificity and clarity in your categorization, steering clear of broad or vague descriptions such as: 'Subject-Specific Knowledge Requirement', 'Task Ambiguity', 'Context-Dependent Jargon Interpretation', 'Specialized Domain Knowledge Required', 'Complex Question Mishandling', 'Ambiguously Phrased Question', 'Implicit Contextual Understanding' 'Implicit Language Competency Assumption', 'Specific Language Processing Tasks', or 'Inference-Based Questions.'
We will evaluate the effectiveness and generalizability of the patterns descriptions using a separate set of unseen questions. Each description should be clear, precise, and comprehensive enough to allow us to reliably identify and categorize new questions that exhibit the same pattern characteristics, even if they differ in their specific content or phrasing.
For each task, please focus on isolating a single task among the questions, rather than grouping multiple tasks under a single task description.
Instead of discussing the content of the questions within each task description, please provide two short snippets of questions that demonstrate the identified task.
The desired format for presenting each task analysis is as follows:
1. **Task Title**: Provide a detailed description of the specific pattern observed in the questions that were answered by the system.
- Example Question 1 (Task 1)
- Example Question 2 (Task 1)

2. **Task Title**: Provide a detailed description of the specific pattern observed in the questions that were answered by the system.
- Example Question 1 (Task 2)
- Example Question 2 (Task 2)

A.2 Prompt for Falsifier

To falsify each category we ask Claude 3 Opus if it can confidently classify as an input as aligned with a category using the prompt below.

Prompt

Before answering, provide your reasoning. Evaluate the alignment between the provided question and the task based on the title and description. Importantly, a question should be considered aligned even if the aspect connecting it to the criteria is not the primary focus of the question. The question isn't required to completely satisfy all the criteria. If any part of the question, no matter how straightforward or simple, meets any aspect of the criteria, it should be considered aligned. Additionally, assess if the task description is too vague, ambiguous, or broadly interpreted to make a confident judgment on alignment and explicitly state the reasons for this uncertainty.
Task Mode: {}
Description of Task Mode: {}
Question: {}
Analysis: Examine the task mode and its description along with the provided question. Identify any elements in the question that relate to the task, regardless of their prominence. Reasoning: Discuss the connections or discrepancies found during the analysis. Explain why these points lead you to consider the question as aligned, unaligned, or too ambiguous to determine. Conclusion: Provide your final assessment based on the analysis, reasoning, and task feedback. If the question aligns with the task, state 'Conclusion: Aligned'. If the question does not align with the task, state 'Conclusion: Unaligned'. If it's unclear whether the question aligns due to vagueness or ambiguity in the task, state 'Conclusion: Uncertain'. If the task is deemed too vague or ambiguous, provide specific reasons and suggest how it might be refined.

A.3 Prompt for Category Scorer

The prompt for scoring categories is similar to the falsification stage, with the exception that it does ask for the confidence of Claude 3 Opus.

Prompt

Before answering, provide your reasoning. Your job is to determine if the question is asking for information described in the task.
Task Mode Title: {}
Description of Task: {}
Question: {}
Instructions:
1. Determine if the question is asking for information or assistance related to the task.
2. Carefully read the question and identify any elements or phrases in the question that are relevant to the task and its description.
3. Conclusion: State 'Conclusion: Yes' if your reasoning identifies a connection between the characteristics described in the task and the characteristics of the question. If, on the other hand, your analysis reveals no such relation, then state 'Conclusion: No'.

B Comparing Claude Across Categories

GSM8K is an English dataset of grade school math questions. We generate categories based on the failure points for Claude 1.2 with zero-shot prompting. GPT-4 is the category proposer and batch size is 60.

For the synthetic question generation, we use few-shot prompting (k=5), selecting examples aligned with each category. Claude 3 Opus generates synthetic questions.

Prompt to generate synthetic questions **Examples of GSM8K questions.**

- Raymond and Samantha are cousins. Raymond was born 6 years before Samantha. Raymond had a son at the age of 23. If Samantha is now 31, how many years ago was Raymond's son born?

Category Descriptions
Elapsed Time Calculation: This pattern deals with questions that require determining the duration or amount of time taken to complete a task, or how long an event lasts, taking into account speed, distance, or a rate of change. Calculations concerning the time involved in back-and-forth or repetitive actions, as well as modifying speeds or intervals, are core to this task.
Predicting Future Values Based on Growth or Change Rates: These questions ask for the determination of a future value given a starting value and a rate of change. This could be a compound interest calculation, a physical growth rate, or an accumulation of value or quantity over time. It requires understanding the rate of change (such as percentage increases or exponential growth) and applying it correctly over the specified time frame.
Temporal and Age-Related Calculations: This task involves calculating the difference in ages between characters or determining an age in the future or past. To complete the task, one must be able to handle relationships and comparisons of ages at different points in time, including calculating a character's age based on another character's age, or their age at different time intervals.
Conversion and Unit Management: These questions involve converting measurements or counts from one unit to another, handling units of time, weight, distance, energy, or currency. They often require the application of conversion factors or aggregation of different units to derive a final value.

Table 8: Full categories descriptions that were discovered based on Claude-1.2 weak points.

Prompt

Construct a question and its solution based on the given task, ensuring the format aligns with the provided examples.
Task: {}
Example Questions and Answers: {}
Your Task: Based on the task that describes certain types of questions, formulate a new question that contains characteristics described in the task and provide an answer following the format of the examples above. Ensure your answers round to a whole number. Ensure the questions you generate are unique.

Figure 4: Synthetic questions prompt that we pass to Claude-3 Opus. The few-shot examples come from questions that are classified as aligned with the hypothesis.

- John had a son James when he was 19. James is now twice as old as his sister Dora, who will turn 12 in 3 years. How old will John's youngest son, who was born when John was 32, be in 3 years?

Examples of synthetic questions Claude 3 Opus generates.

- Maya is 12 years younger than her brother, who is half the age of their father. If the sum of their current ages is 58, how old will Maya be when her brother is their father's current age?

- Tanya is 24 years old. Her daughter Lily is one-third her age. If Tanya's mother Carol is 30 years older than Tanya, how old will Lily be when Carol turns 90?

Category	Origin	Semantic	Diff. Rewrite	Average
Circular Shifting of DataFrame Columns	0.0% $^{-36.0}$	30.0% $^{-0.2}$	0.0% $^{-10.0}$	12.5% $^{-13.7}$
Splitting and Expanding Lists in Cells	20.0% $^{-16.0}$	12.5% $^{-17.7}$	25.0% $^{+15.0}$	18.2% $^{-8.0}$
Filtering Overlapping Dates	0.0% $^{-36.0}$	0.0% $^{-30.2}$	0.0% $^{-10.0}$	0.0% $^{-26.2}$
Identifying and Handling Duplicate Rows in Pandas DataFrames	12.5% $^{-23.5}$	27.8% $^{-2.4}$	5.9% $^{-4.1}$	15.7% $^{-10.5}$
DS-1000 Baseline	36.0%	30.2%	10.0%	26.2%

Table 9: Codex-002 accuracy on the Pandas subset of DS-1000. Where "Diff. Rewrite" is short for Difficult-Rewrite. The performance of Codex-002 drops considerably on the categories ASICS generates. The values in the upper right corner of each row in the score column represent the deviation from the baseline performance.

C Finding Categories in Human Preferences

C.1 Finding Categories in Data Science Inputs

DS-1000 is a code generation dataset encompassing problems from seven libraries, including Pandas, Numpy, and PyTorch. Created from StackOverflow questions, the dataset includes perturbed questions to prevent models from recalling solutions from pre-training. Semantic perturbations modify the question's wording without changing its difficulty, while difficult rewrites make the questions more complex and challenging. We evaluate Codex-002 on the Pandas subset of DS-1000, as it contains the most questions of all the libraries in the dataset. We use the Codex-002 results reported by Lai et al. (2022) to obtain $\mathcal{Q}$, which we pass into ASICS. We use Claude 3 Opus as the *category proposer*.

In Table 9 we present the categories ASICS generates and their scores. The scores for each category are considerably less than the baseline performance, for example, Codex-002s performance drops by 26% on questions related "Filtering Overlapping Dates." Lai et al. (2022) perform an error analysis of Codex-002 on DS-1000 and find that it struggles with questions that require multi-step reasoning. Our findings agree with theirs but ASICS produces these failures autonomously and provides more fine-grained categories such as "Circular Shifting of Dataframes", and uncovers new weak points, like "Identifying and Handling Duplicate Rows in Pandas DataFrames".

Experimental Details We generate task descriptions based on 1000 prompts where humans prefer alternative models to LLaMA-3. We generate categories with batch size n=50. GPT-4 is the category proposer.

Additional Hypothesis Testing Details. For scores greater than the baseline score, a right-sided test was used. For scores less than the baseline proportion, a left-sided test was used.

Category
Strategic Problem-Solving in Logic Puzzles: Solving riddles or logic puzzles that require a methodical approach to find the most efficient solution, often involving a step-by-step process of elimination or strategic planning, without superfluous information that could complicate the puzzle.
Management Advice for Specific Health Issue: This task requires empathetically providing practical and possibly medically-informed suggestions or tips for managing a specific health-related issue. The response should convey understanding while presenting actionable and relevant advice considering the health context described.
Generating Pseudoprofound Content: The task requires constructing a statement or piece of text that carries an appearance of profundity or deep insight while lacking substantive information or clarity. Such statements often employ ambiguous phrasing with a semblance of philosophical or poetic depth.
Abstract Spatial Problem Solving: The task involves solving abstract or visual spatial problems that require deducing relative positions or movements based on described scenarios. This often includes mental manipulation of geometric figures or directional movements.
Conceptual Explanation in Mathematics or Physics: This task requires a clear understanding of advanced concepts in mathematics or physics to elucidate why a mathematical expression or a physical phenomenon behaves in a particular manner, necessitating an ability to translate complex, abstract concepts into accessible, accurate explanations.

Table 10: Full description of the categories shown in **??**.

D Finding Categories in Data Science Questions

DS-1000 is an English dataset of data science questions. Claude 3 Opus is the category generator. We set batch size to be 60. There are more than 5 questions aligned with each category we identified. The full descriptions for the categories are shown in Table 11.

Category
Circular Shifting of DataFrame Columns: The model has difficulty with questions that require circularly shifting the values in a specific DataFrame column.
Splitting and Expanding Lists in Cells: The questions ask to split lists contained within dataframe cells into separate columns.
Filtering Overlapping Dates: The questions require filtering out rows where dates are within a certain time period of each other.
Identifying and Handling Duplicate Rows in Pandas DataFrames: The model has difficulty with questions that involve identifying duplicate rows in a DataFrame based on specific columns and performing actions on the duplicates, such as adding new columns or keeping specific instances of the duplicates.

Table 11: Full descriptions of categories ASICSfound based on the DS-1000 dataset.

E Validating Automatic Hypothesis Validation Method

Questionnaire format.

- **Hypothesis:** The title and description of the hypothesis under evaluation.
- **Option A:** A question randomly sampled from either the aligned or unaligned dataset.
- **Option B:** A question randomly sampled from the opposite dataset (aligned or unaligned) compared to Option A.

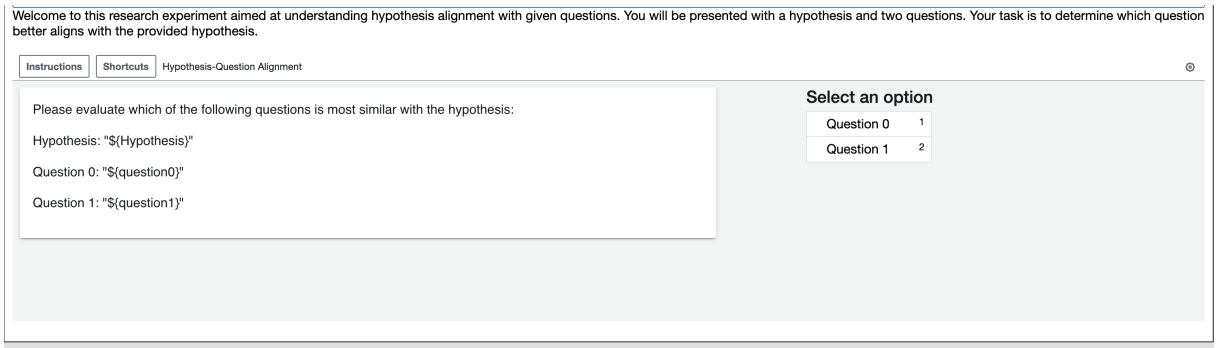

Figure 5: The amazon mechanical turk interface our annotators are exposed to while they provide feedback.

Crowdworker Compensation. To adequately compensate our annotators, we studied how long they would require to perform our tasks. For the first round, our preliminary experiments showed that each worker would need approximately 5 seconds to annotate a category, question pair. Assuming Turkers worked for an hour, they would be paid 21.60 USD which is above the US federal minimum wage. Based on this, we decided to pay our workers who were based on the US East Coast $0.03/task. The hypotheses that our crowdworkers evaluated are shown in Table 13.

Categories
Chronological Event Sequencing: The task involves ordering a series of events within a narrative or historical context. The task requires an understanding of temporal cues and the ability to organize pieces of information in a sequence that respects the chronological progression of the narrative or actual historical timeline..
Riddle Resolution Based on Textually Provided Information: The model failed at resolving riddles when the solution relies solely on interpreting the information that is explicitly stated in the question without requiring additional external knowledge. The resolution of these riddles depends on analyzing the given descriptive scenario, identifying relevant pieces, and applying logical reasoning to deduce the answer directly from the details provided.
Creative Literary Editing and Variation: The task involves editing and improving a piece of narrative text by enhancing its sentence flow, word choice, and vivid descriptions. It requires understanding the original tone and style, and applying enhancements without changing the core narrative.
Abstract Spatial Problem Solving: The task involves solving abstract or visual spatial problems that require deducing relative positions or movements based on described scenarios. This often includes mental manipulation of geometric figures or directional movements.
Creative Character Self-Depiction: The task involves choosing between two options to describe oneself in a way that aligns with the characteristics or themes of those options, which are often from pop culture references or creative prompts requiring a degree of personal reflection or creative interpretation.

Categories
Problem-Solving with Familial Relationship Dynamics: This task involves solving problems based on descriptions of familial relationships and dynamics that require logical deduction and a clear understanding of relationships such as siblings, parents, and in-laws.
Advanced Relationship Dynamics and Social Cues Interpretation: This involves questions that hinge on the ability to discern complex social interactions, often requiring nuanced understanding of human behavior, unspoken social cues, or emotional undercurrents that are commonly understood by humans but not explicitly stated in the context provided.
Conceptual Explanation in Mathematics or Physics: This task requires a clear understanding of advanced concepts in mathematics or physics to elucidate why a mathematical expression or a physical phenomenon behaves in a particular manner, necessitating an ability to translate complex, abstract concepts into accessible, accurate explanations.

Table 12: Human preference categories evaluated by human annotators.

GSM8K evaluation The categories generated for the GSM8K variant of the human evaluation were generated with a slightly different category proposer prompt. The different category proposer and scorer prompts are shown in Appendix F. We find that Claude-3 Opus is able to generate results aligned with human opinions across different variants of the same prompt. The categories the volunteers expert annotate are shown in Table 14.

Table 13: Randomly sampled categories evaluated via crowdwsourcing.

Erroneous Application of Percentage Changes: The model does not correctly apply percentage changes over multiple transactions or time periods. This includes incorrectly calculating the results of consecutive percentage increases or decreases or misapplying compound percentage changes.
Considering Temporal Aspects in Computation Failure: This failure mode is characterized by the model's inability to correctly process and calculate numerical values that are dependent on the passage of time, for example, rates per hour, growth per day, or accumulation in weeks. The error arises when the model needs to multiply or divide quantities by factors related to a temporal dimension to determine the final figures and overlooks the sequential nature of time-dependent growth or reduction.
Multi-Scenario Financial Computation Breakdown: Challenges arise when the model faces questions involving financial scenarios that require multiple transactional computations or when the question involves multiple distinct stages of financial transactions (e.g., initial investment, interest or profit accrual, deductions, and final net computations). The failure is noticeable when the problem presents a scenario where initial values are influenced by subsequent financial events, or there are layered transactions to be tracked.

Table 14: Categories evaluated by expert volunteers.

GSM8K Failure Modes
Complex Conditional Probability Misinterpretation: The model struggles with problems that involve conditional probabilities with multiple conditions or layers, especially when the problem requires combining probabilities from different events or understanding the subtleties of dependent and independent probabilities.
Failure in Addressing Compound Growth and Decay: The model does not appropriately handle problems that require understanding and calculating compound growth or decay rates, especially when the growth or decay is non-linear and incremental changes apply to the previous amount rather than the initial value
Erroneous Application of Percentage Changes: The model does not correctly apply percentage changes over multiple transactions or time periods. This includes incorrectly calculating the results of consecutive percentage increases or decreases or misapplying compound percentage changes.
Time and Rate Problem Misinterpretation: The model inaccurately solves problems involving time and rate, particularly when these must be translated into distance or work done. This typically happens when the scenario requires converting units or applying the rate-time-distance relationship in reverse.
Considering Temporal Aspects in Computation Failure: This failure mode is characterized by the model's inability to correctly process and calculate numerical values that are dependent on the passage of time, for example, rates per hour, growth per day, or accumulation in weeks. The error arises when the model needs to multiply or divide quantities by factors related to a temporal dimension to determine the final figures and overlooks the sequential nature of time-dependent growth or reduction.
Discount Application Error: The model demonstrates difficulty when it must apply a percentage-based discount or markup to multiple items. The model's mistakes in this failure mode include incorrectly calculating the total after applying the discount or failing to subtract the discount from the original total. This failure occurs with questions that require the application of a percentage to adjust the total cost or value of individual or groups of items.
Exponential Growth Calculation Failure: The model is unable to accurately perform calculations that involve exponential growth or compound processes, such as repeated multiplication over time. This is characterized by the model's inability to iterate a multiplication process over several periods or stages. The failure occurs when the situation demands iterative multiplicative increases or compounding effects, as is common in scenarios involving interest rates, population growth, or disease spread.
Multi-Scenario Financial Computation Breakdown: Challenges arise when the model faces questions involving financial scenarios that require multiple transactional computations or when the question involves multiple distinct stages of financial transactions (e.g., initial investment, interest or profit accrual, deductions, and final net computations). The failure is noticeable when the problem presents a scenario where initial values are influenced by subsequent financial events, or there are layered transactions to be tracked.
Cumulative Scenario Comprehension Slip: The model has difficulties correctly answering questions involving cumulative scenarios where one must keep track of an ongoing tally, such as remaining or leftover items or the accumulation of goods or deficits over time. The issue becomes apparent in questions where there is a need to add to and subtract from a running total continually. The failure arises when it is required to maintain a tally through multiple steps, which often involves both addition and subtraction.

F Prompt Variants

Additional Proposer Prompt

Prompt

I will provide a series of data for you to remember. Subsequently, I will ask you some questions to test your performance! Here are questions for you to memorize. n individual questions The above questions were incorrectly answered by the machine learning model. Using these specific examples, are there any common features or general types of failures the model is making? For each identified pattern of error, please offer a detailed and precise description. Highlight the particular characteristics of the questions that the model failed to answer correctly. Please focus on achieving a high level of specificity and clarity in your categorization, steering clear of broad or vague classifications such as 'Poor Implicit Information Understanding', 'Lack of Real-World Context or Understanding', 'Misunderstanding of Natural Language', 'Complex Question Mishandling', 'Incorrect Interpretation of the Problem', or 'Inadequate Understanding of Contextual Concepts.' For each failure, please focus on isolating a single issue, idea, or feature that leads to a failure, rather than grouping multiple issues under a single failure mode. These error patterns will be verified against unseen data from the same distribution, necessitating clear, concise, and specific hypotheses for accurate evaluation. It is crucial to articulate these hypotheses clearly and concisely. Instead of discussing the content of the questions within each failure mode description, please provide two short snippets of questions that demonstrate the identified failure mode afterward. This format ensures a clear distinction between the analysis and the examples. Precisely describe the specific conditions for when the failure occurs. The desired format for presenting each failure mode analysis is as follows:
Failure Mode Title: A detailed description of the failure mode. Specify the conditions under which the failure occurs.
Example Question 1 (Demonstrating Failure Mode 1)
Example Question 2 (Demonstrating Failure Mode 1)
Failure Mode Title: A detailed description of the failure mode. Specify the conditions under which the failure occurs.
Example Question 1 (Demonstrating Failure Mode 2)
Example Question 2 (Demonstrating Failure Mode 2)

Additional Category Scorer Prompt

Prompt

Before answering, provide your reasoning. Please evaluate the alignment between the provided question, the failure mode title, and its description. Your task is to determine if the question is aligned with the failure mode and its description. The question is not required to completely align with the entire hypothesis description. If any part of the question, no matter how straightforward or simple, somewhat meets the description, it should be considered aligned within the proposed failure mode. Importantly, a question should be considered aligned even if the aspect connecting it to the failure mode title and description is not the primary focus of the question.
Failure Mode Title: {}
Description of Failure Mode: {}
Question: {}
Instructions:
Analyze each point of the failure description presented. Carefully read the question and identify any elements or phrases in the question that are relevant to the failure mode and its description.
Conclusion: State 'Conclusion: Yes' if your reasoning ascertains any connection to the failure mode and its description within the question content. If, on the other hand, your analysis reveals no such relation, then state 'Conclusion: No'.