PromptScore: Evaluating Large Language Model Performance using a Comprehensive Prompt Scoring System

Amritansh Mishra amritanshmis

Manish Ranjan Karna mkarna

Manas Wadhwa mwadhwa

Rahul Saxena rahulsaxena

1 Problem statement

Our project aims to develop a mechanism to measure the specificity of prompts used to interact with Large Language Models (LLMs) like GPT-4. We're creating a scoring metric to evaluate the effectiveness of prompts based on criteria like coherence, clarity, and ambiguity. The goal is to establish a standardized ”prompt score” validated by human annotators, which reflects the specificity of a prompt. To obtain the prompt score, we tried two approaches. First we fine tuned Llama (Touvron et al., 2023) model using QLoRA (Dettmers et al., 2023) technique on our dataset. Second we used GPT-4 to generate the scores based on few shot prompting strategy. We compare the results in the following sections. This score will enable researchers to assess LLM performance based on prompt specificity, leading to a more nuanced understanding of LLM capabilities.

We believe a standardized prompt scoring system will significantly advance the field of LLMs and have practical implications. Reliable prompt scoring will allow users to tailor LLMs to their specific needs, enhancing their utility and performance in various applications. In this report, we examine how certain LLMs (Alpaca, Claude, and Gemma) struggle with generating responses to highly specific prompts, and evaluate their performance on coherence, constraint following ability, and fluency metrics. We did this comparison by taking prompts in the order of increasing prompt specificity (prompt score) and evaluate different LLMs.

Our hypothesis claims that as specificity increases there are much less data for the LLM to train with which would eventually hinder LLM performance.

2 What you proposed vs. what you accomplished

We proposed to develop a mechanism to measure the specificity of prompts used to interact with Large Language Models (LLMs) like GPT-4. This is a scoring metric to evaluate the effectiveness of prompts based on criteria like coherence, clarity, and ambiguity. Our goal was to establish a standardized ”prompt score” validated by human annotators, which reflects the specificity of a prompt, and use this prompt scoring mechanism to compare various LLMs and assess their performance based on prompt specificity. We have accomplished all of the milestones we set out to achieve. We developed two approaches to obtain the prompt score. Our results provide valuable insights into the performance of LLMs and the effectiveness of our prompt scoring mechanism.

3 Related work

The development and evaluation of prompt specificity in large language models (LLMs) has been an area of significant research interest with great potential. Our project builds on a substantial body of research focused on prompt design and evaluation in LLMs. This section contextualizes our work within the broader literature, highlighting key studies and how our approach extends or differs from existing methods.

We leverage the seminal work of (Wei et al., 2022), ”Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” which demonstrated that using chain-of-thought prompting can significantly improve a model's performance on tasks like arithmetic, commonsense reasoning, and other complex tasks. We incorporated this idea into our project while evaluating the generated responses using GPT-4 to ensure thorough and logical evalution.

Similarly, (Liu et al., 2023) in ”G-EVAL: NLG Evaluation using GPT-4 with Better Human Alignment” focused on evaluating natural language generation (NLG) systems. They found that conventional reference-based metrics like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) have relatively low correlation with human judgments. They proposed using LLMs as reference-free metrics for NLG evaluation, which showed better alignment with human judgments. Inspired by this, we used GPT-4 to judge the quality of prompt responses (in this case, stories) generated by other models like Alpaca and Claude. By developing a similar scoring metric, we aimed to evaluate the responses of various LLMs more effectively, focusing on metrics like coherence, consistency, and fluency.

Our approach to evaluating the instruction-following ability of LLMs like Claude and Gemma was inspired by (Qin et al., 2024) ”INFOBENCH: Evaluating Instruction Following Ability in Large Language Models.” Their findings demonstrated that GPT-4 can serve as a cost-efficient annotator. We utilized GPT-4 as an annotator (along with human annotations) while generating the dataset of prompts used for fine-tuning the Llama model.

Additionally, we drew inspiration from (Zhou et al., 2023b).'s ”Instruction Following Evaluation of Large Language Models.” We adopted their idea of focusing on a set of ”verifiable instructions,” such as ”write in 400 words” and including some specific keywords to evaluate the constraint-following ability of LLM generations.

Recent studies have underscored the significance of data quality in human annotations, which is crucial for ensuring that large language models (LLMs) adhere closely to instructions, thereby minimizing discrepancies between the generated outputs and the users' intended inputs (Zhou et al., 2023a); (Köpf et al., 2023). Consequently, we prioritized human annotations in the preparation of our prompts dataset. This approach allowed for a more accurate fine-tuning of the LLaMa model, specifically tailoring it to effectively relate the prompts to various metrics such as the number of constraints and clarity. Additionally, we explored techniques to enhance the instruction-following capabilities of LLMs, notably through the method of instruction backtranslation, as discussed in prior research (Li et al., 2024).

In the process of generating a story prompt dataset, it is critical to encompass a broad spectrum of themes and scenarios. This approach ensures a balanced knowledge base from which the fine-tuned model can learn. The work of (Kandpal et al., 2023) on ”Large Language Models Struggle to Learn Long-Tail Knowledge” reveals that the performance of large language models (LLMs) on knowledge-based tasks is significantly influenced by the prevalence of related information in their training datasets . By diversifying the content of the story prompts, we mitigate the risk of a skewed representation, where certain types of information or prompt categories are underrepresented—a problem commonly referred to as the long tail issue. Such diversity in training data is essential for developing models capable of handling and understanding a wide variety of inputs and maintaining robustness in interpreting how text relates to constraints, clarity, and other metrics. This is further corroborated by the work of (Ravichander et al., 2020), which shows that language models make inconsistent predictions when prompted with similar statements.

4 Dataset

We created our own dataset consisting of 800 prompts and their annotated scores. We score each prompt based on four criteria: number of constraints, constraint complexity, clarity, and prompt complexity. Each criterion is scored on an integer scale from 1 to 5. The dataset is used later to fine-tune a LLM so that it can learn to give scores to each prompt.

Dataset Statistics

- Size: 800 examples
- Source: Online websites (blog.reedsy.com), ChatGPT, and team-written prompts
- Domain: Story domain

Below are three prompts, with increasing specificity, along with their corresponding annotated values.

4.1 Data preprocessing

We didn't perform extensive data preprocessing as most prompts were already clean and required minimal cleaning.

4.2 Data annotation

We divided the dataset among team members for annotation. The main challenges we faced were

Instruction: Evaluate the following prompts based on the given criteria.

1. **Prompt:** "Write a story about an educational institution that doesn't provide education."
   - **#constraints:** 2
   - **Constraint Complexity:** 2
   - **Clarity:** 5
   - **Prompt Complexity:** 1
2. **Prompt:** "Write a short story with an unreliable narrator that readers can never quite trust. In the first sentence of your story, have the narrator tell us exactly what's going to happen — but not how it will happen."
   - **#constraints:** 3
   - **Constraint Complexity:** 3
   - **Clarity:** 5
   - **Prompt Complexity:** 3
3. **Prompt:** "Write a story with the following constraint: Title: The Forgotten Path Each paragraph starts with "Lost in". Total of 5 paragraphs. Each paragraph consists of exactly 7 sentences. All sentences have an even number of words. Total word count: 140-150 words. Must include the phrase "whispering winds". Ending line: "And so they ventured on, into the unknown."
   - **#constraints:** 5
   - **Constraint Complexity:** 5
   - **Clarity:** 4
   - **Prompt Complexity:** 5

Figure 1: Prompt Examples

assigning similar values (e.g., clarity scores of 4 or 5) and the limitations of the 'Clarity' feature, which had similar scores for most examples. Thus 'Clarity' feature did not help much in prompt score as it was almost same for all the examples. Despite these issues, team members generally agreed on annotated values, with minimal disagreements.

4.3 Criteria Guidelines

Constraints Constraints are rated based on their quantity and complexity in the problem. A rating of 1 indicates there are hardly any constraints, while a rating of 2 suggests the presence of one or two constraints. A rating of 3 is given when there are between two and three constraints, rating 4 for three to five constraints, and a rating of 5 is assigned when there are more than five constraints.

Constraint Complexity The complexity of constraints also significantly influences their rating. Constraints are considered very simple, warranting a rating of 1. If one or two constraints are tricky, but the rest remain simple, a rating of 2 is appropriate. A moderate difficulty level in constraints earns a rating of 3. Constraints that include tricky elements, such as incorporating specific words with certain frequencies, receive a rating of 4. The highest complexity, which involves multiple tricky constraints, is rated at 5.

Clarity Clarity in the presentation of constraints and prompts is critical. Ratings of 1 or 2 are given for very unclear guidelines. An ambiguous presentation results in a rating of 3, while very clear instructions receive a rating between 4 and 5.

Prompt Complexity Prompt complexity refers to the intricacy of the main storyline or the outline of how the story should be structured. Simple and straightforward story outlines receive a rating of 1 or 2. If the storyline is somewhat trivial but involves minor complexities, it is rated at 3. More complex storylines, particularly those involving less well-known topics or intricate plots, are rated between 4 and 5.

5 Baselines

In our innovative study, we compared two primary models, LLama and GPT-4, to evaluate the effectiveness of automated prompt scoring in generating and assessing story prompts. Since PromptScore has never been evaluated before we don't have a baseline to compare.

5.1 Llama Fine-Tuning

(Touvron et al., 2023) Llama, a transformer-based language model, was fine-tuned using a specialized dataset comprising story prompts and evaluations. We opted for a learning rate of 2e-5 after experimenting with various rates, finding it optimal for our purposes. The training process involved batches of 16 over 15 epochs, utilizing the AdamW optimizer for improved weight handling. Despite these adjustments, LLama's performance remained suboptimal in comparison to GPT-4, particularly struggling with the nuanced evaluation criteria and maintaining coherence in prompt generation.

5.2 GPT-4 Few-Shot Learning

Contrastingly, GPT-4 was employed in a few-shot learning configuration, where it was provided with

Hyperparameter	Value
Learning Rate	2e-5
Batch Size	16
Epochs	15
Optimizer	AdamW

Table 1: Hyperparameters used in fine-tuning the Llama model

a handful of example prompts to guide its generation and evaluation processes. We determined that 200 example shots were the most effective in guiding GPT-4 without overwhelming it. In tests, GPT-4 significantly outperformed LLama, demonstrating superior understanding of the evaluation criteria and generating more coherent, high-quality prompts.

5.3 Dataset Splits

To ensure a robust evaluation, the dataset was split into 70% for training, 15% for validation, and 15% for final testing. This division was strategically chosen to maximize learning while providing sufficient data for validation and unbiased testing. Importantly, no hyperparameter tuning was conducted using the test set to prevent data leakage and ensure the integrity of our results.

This study underscores the potential of GPT-4 in automating the flow of prompt scoring, leading us to select it over LLama for its superior performance and reliability in handling complex evaluation tasks.

6 Methodology

6.1 Calculating Prompt Score

To evaluate each prompt in our curated dataset, we score them based on four criteria: the number of constraints, the complexity of individual constraints, the clarity of the prompt, and the overall prompt complexity. The definitions of these criteria are as follows:

Number of Constraints: The total number of constraints in the prompt. A higher number of constraints typically makes a prompt more specific.

Constraints Complexity: The complexity of each individual constraint. Prompt Complexity: The complexity of the overall prompt. A prompt can be complex even if individual constraints are simple. For example, ”Write a story about a man with four legs in 100 words” has only two constraints (four legs and 100 words), but the prompt is complex due to the unusual nature of the story and the brevity required.

Clarity: How clear and understandable the prompt is. To ensure accurate scoring, we provide well-defined guidelines for annotators. A high prompt score indicates a high level of specificity.

Prompt Complexity refers to how high or how complex the base prompt is. For example a prompt like ’A boy born in Delhi’ would be much less complex ’Come up with a story of a man born with four legs’.

”Come up with a story which has seven characters and a place called Bhopal, where the story revolves around children betraying parents.”

For this prompt, we assign scores based on several criteria as follows:

- **Number of Constraints**: 3
- **Constraints Complexity**: 2
- **Clarity**: 5
- **Prompt Complexity**: 2

Using an annotated dataset, we fine-tune the LLaMA2 model using the QLoRA approach. As all prompts received near-perfect clarity scores, we exclude this criterion during fine-tuning to prevent biasing the model. We enhance the model by adding an MLP layer atop the final layer’s last hidden state representation to predict three values: the number of constraints, constraints complexity, and prompt complexity.

When assessing new prompts, we employ both our fine-tuned model and few-shot prompting with the ChatGPT-4 model. Our findings reveal that the prompt scores generated by ChatGPT-4 are superior to those produced by our fine-tuned model, likely attributable to the vast training data available to ChatGPT-4, in contrast to more limited datasets accessible for models like LLaMA2.

6.2 Generating Prompts for Story Evaluation

Once we have designed a metric to effectively evaluate the specificity of prompts, we proceed to iteratively refine these prompts to cover a range of specificities. The process involves the following steps:

To effectively evaluate the specificity of prompts, we follow an iterative refinement process. First, we begin with a broad, general prompt to set the foundation for story evaluation. Next, we incrementally add constraints to this initial prompt using a Large Language Model (LLM). Each iteration involves asking the LLM to add one or more specific elements or requirements to the existing prompt, thereby increasing its specificity. Constraints to add include the number of paragraphs, word count in each paragraph, the theme of each paragraph, words to add with frequency, and ending statements or a specific line. For example, an initial prompt might be, ”Write a short story about a journey taken by a young girl.” We repeat this process of adding constraints until we achieve a series of prompts with varying levels of specificity, aiming to generate a total of 10 prompts, each more specific than the previous one. The results are stored in a database to avoid making repeated API calls to the LLM, as this increases the latency of the code.

6.3 Evaluation of Story Generated

In our study, we employed GPT-4 as an evaluator to assess the quality of stories generated from prompts by various language models (LLMs). Drawing inspiration from the methodology outlined in the paper (Liu et al., 2023), we structured our evaluation process to systematically measure the performance of each LLM on several key criteria. The evaluation process was meticulously designed to ensure a comprehensive and objective assessment.

6.3.1 Evaluation Instructions

We provided GPT-4 with detailed instructions to evaluate the generated stories. The instructions were formatted as a prompt and included specific guidelines for assessing each story based on the following categories:

- **Coherence**: This criterion measures the logical consistency and narrative flow of the story. A coherent story maintains a clear and understandable progression of events, with well-connected ideas and actions.

- **Constraints Satisfied**: This criterion evaluates how well the generated story adheres to the specific constraints or requirements outlined in the prompt. These constraints might include specific characters, settings, themes, or plot elements that the story must incorporate.

- **Fluency**: This criterion assesses the overall readability and linguistic quality of the story. A fluent story is grammatically correct, well-punctuated, and free from awkward phrasing or unnatural language use.

6.3.2 Evaluation Process

To conduct the evaluation, we followed these steps:

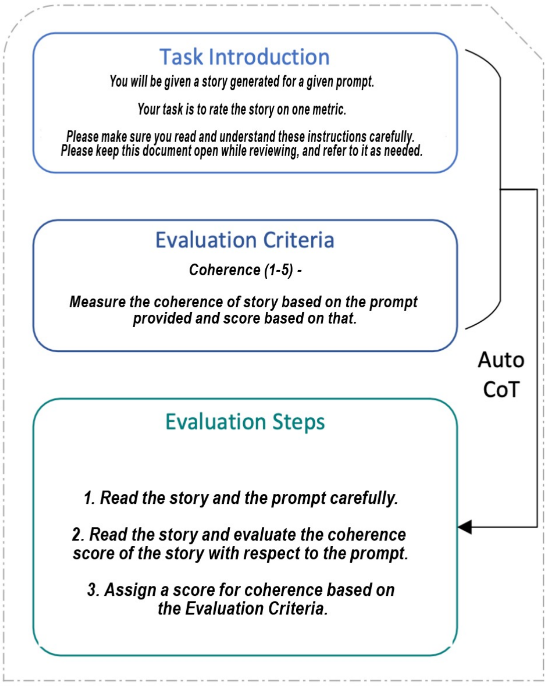
1. **Prompt Design**: We designed prompts that included both the story prompt and the evaluation instructions for GPT-4. The evaluation instructions were formatted in a clear and concise manner to ensure that GPT-4 could effectively assess each story based on the specified criteria. The prompts were constructed to guide GPT-4 to follow a structured reasoning process, known as Chain of Thought (CoT), which helps in breaking down the evaluation into logical steps.

2. **Story Generation**: Multiple LLMs were used to generate stories based on the same set of prompts. This allowed us to compare the performance of each model on a common basis. By generating multiple stories, we could analyze the diversity and quality of outputs from different models under consistent conditions.

3. **Evaluation by GPT-4**: For each generated story, GPT-4 was tasked with providing an evaluation based on three criteria: Coherence, Constraints Satisfied, and Fluency. The evaluations were conducted independently for each story to maintain objectivity. We passed the instructions to GPT-4 as a prompt along with the story it had to evaluate. The CoT reasoning was encouraged to ensure thorough and logical evaluation.

   An example of how instructions are passed to GPT is presented in the figure 2

4. **Scoring and Analysis**: The scores provided by GPT-4 were then compiled and analyzed to determine the relative performance of each LLM. The analysis focused on identifying strengths and weaknesses in the generated stories and understanding how each model performed across different evaluation categories.


Task Introduction

You will be given a story generated for a given prompt.

Your task is to rate the story on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria

Coherence (1-5) -

Measure the coherence of story based on the prompt provided and score based on that.

Evaluation Steps

1. Read the story and the prompt carefully.

2. Read the story and evaluate the coherence score of the story with respect to the prompt.

3. Assign a score for coherence based on the Evaluation Criteria.

Figure 2: Passed Instructions to GPT

We managed to complete a working implementation using libraries such as PyTorch, Huggingface, Trainer, and others. We did not rely on any existing implementations and instead implemented our own models. Specifically, we implemented the Llama model, with the associated files being `Llama` and `trainer`. Our experiments were conducted on GPU machines, which provided the necessary computational power for training and evaluation. There were no significant issues that we could not solve during the implementation process. Additionally, we did not need to employ any Colab-specific hacks for training our model, as our setup and resources were sufficient for our needs.

7 Results

We tested our prompt score algorithm on a series of 10 prompts with increasing specificity, plotting the scores in a graph. The results validate our assumption that as specificity increases, the prompt score increases. While the graph shows some fluctuations, potentially due to model errors, the overall trend supports our hypothesis. To obtain the 10 prompts, we started with a simple story writing prompt and incrementally added various constraints using GPT. We manually provided a set of constraints, including parameters such as the number of paragraphs, word count per paragraph, theme, required words with frequency, and specific ending statements. By gradually adding these constraints, we created a range of prompts with increasing specificity, allowing us to evaluate our algorithm's performance.

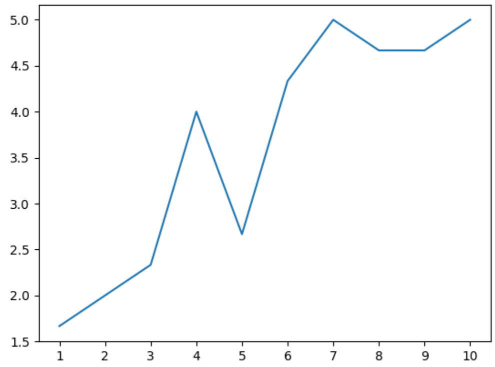


Figure 3: Prompt score for 10 increasingly specific prompts

Building on our successful development of a prompt scoring mechanism, we proceeded to evaluate the performance of three Large Language Models (LLMs) - Alpaca, Gemma, and Claude - on various metrics. Our goal was to assess which model excels in specific areas, namely coherence, constraints following ability, and fluency. To achieve this, we tested each metric individually for all three models, analyzing their performance on prompts with increasing specificity. The following three graphs illustrate the results of this evaluation, providing valuable insights into the strengths and weaknesses of each LLM. By comparing their performance on these key metrics, we can better understand how our prompt scoring mechanism can be used to optimize LLM performance and improve user experience. This is shown in 4, 5 and 6.

A closer examination of the coherence and constraints graphs reveals that Gemma most closely aligns with our hypothesis, followed by Claude, while Alpaca struggles to meet our expectations. Notably, the fluency graphs for all three models exhibit identical trends, indicating that each model's responses are equally fluent. To further investigate the coherence and constraints following ability, we opted to replot the graphs with prompts numbered from 1 to 10 on the horizontal axis, increasing in specificity. This adjust-

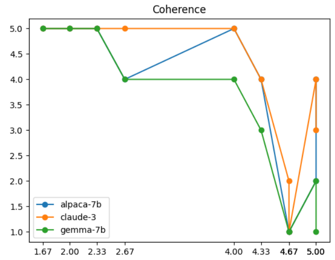


Figure 4: Compare coherence

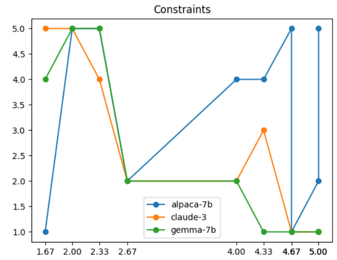


Figure 5: Compare constraints

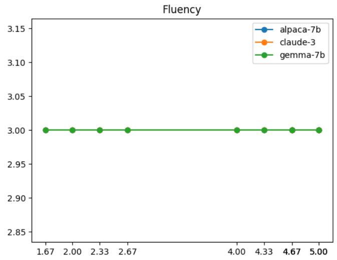


Figure 6: Compare fluency

ment enables a more nuanced visual analysis of the decreasing trend, as the previous graphs had scores on the horizontal axis that weren't uniformly spaced between consecutive prompts. By replotting the graphs in this manner, we can more effectively assess how each model's performance deteriorates as prompt specificity increases, providing valuable insights into their strengths and weaknesses. The resulting graphs offer a clearer visualization of the models' performance, allowing us to better understand their capabilities. This is shown in graphs 7 and 8.

To gain a comprehensive understanding of the three models' performance, we analyzed the average performance of Alpaca, Gemma, and Claude across all three constraints - Coherence, Constraints following ability, and Fluency - using the same set of 10 prompts. Our hypothesis posits that as prompt specificity increases, the models' performance should deteriorate, reflecting their struggle to provide coherent output. The graphs indeed validate this hypothesis, with Claude exhibiting the most pronounced decreasing trend, indicating its ability to verify our hypothesis. Gemma also displays a similar behavior, albeit with some variation. However, Alpaca's performance deviates from the expected trend, failing to demonstrate a clear decline in performance as prompt specificity increases. This disparity in performance among the three models offers valuable insights into their strengths and weaknesses, enabling us to refine our prompt scoring mechanism and optimize LLM performance. This also provides some clue that for random prompt, to judge its performance we can use the Claude model. These graphs are shown in 9, 10, 11.

8 Error analysis

While our prompt scoring mechanism has shown promising results, we observed some errors and limitations in its performance. Specifically, we noticed that for extremely specific prompts, the model's scores on the 'constraints' and 'prompt' complexity metrics were lower than expected. For instance, consider a prompt with very specific requirements as given below.

```
Write a story with the following
   constraints:
- Title: The Enchanted Forest
- Each paragraph starts with "In
   ".
- Total of 5 paragraphs.
- Each paragraph consists of
   exactly 4 sentences.
- Total word count: 100-120 words
   .
- Cannot use the word "magic".
- Must include the phrase "
   whispers of the trees".
- Must have a character named "
   Evelyn".
- Ending line: "And the forest
   was never the same again."
```

The model predicted scores of 4 and 3 for 'constraints' and 'prompt' complexity, respectively. However, according to our human-annotated training data, such highly specific prompts would typically receive scores of 5 and 5. This discrepancy suggests that the model may struggle with overly specific prompts, potentially due to limited training data. This error highlights the need for further refinement and expansion of our training dataset to better capture the nuances of highly specific prompts and improve the model's performance in these cases. This error could also happen since as of now we're predicting integer scores between 1 to 5, so the model might be confused between 4 and 5. Had this been a continuous score, we could have got scores which is more towards what human would score.

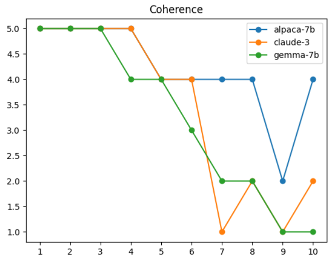


Figure 7: Coherence performance

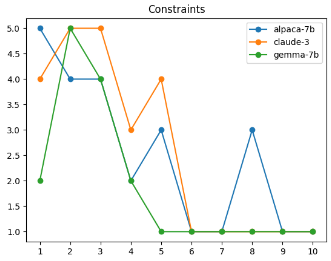


Figure 8: Constraints performance

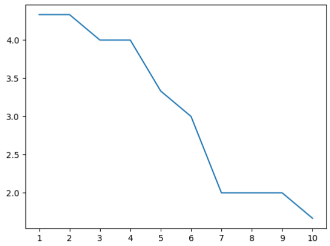


Figure 9: Claude overall performance

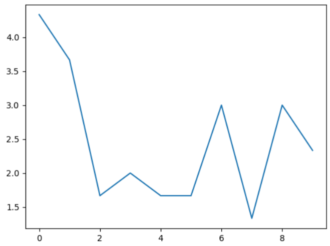


Figure 10: Alpaca overall performance

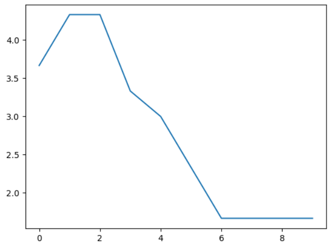


Figure 11: Gemma overall performance

9 Contributions of group members

- Amritansh Mishra: Data preparation and annotation, build training models, database integration, fine tuning, evaluation pipeline.

- Manish Ranjan Karna: Data preparation and annotation, fine tuning, models integration, performance analysis

- Manas Wadhwa: Data preparation and annotation, error analysis, report writing, fine tuning, story generation pipeline.

- Rahul Saxena: Data preparation and annotation, models integration, report writing, story generation pipeline.

10 Conclusion

From our analysis, we notice a clear pattern in the performance of the three models across different metrics. Specifically, when it comes to coherence, Gemma outperforms the others, followed closely by Claude, while Alpaca struggles to maintain coherence. Similarly, in terms of constraints following ability, Claude excels, with Gemma coming in second and Alpaca trailing behind. Furthermore, the individual performance graphs reveal a consistent trend: Claude consistently delivers the best performance, followed by Gemma, and then Alpaca. This clear hierarchy of performance suggests that Claude is the most effective model at handling increasingly specific prompts, while Alpaca faces significant challenges in this regard. Therefore, we can conclude that Claude is the top-performing model, followed by Gemma, and Alpaca requires further refinement to meet the standards set by its peers. We also fine tuned (Touvron et al., 2023) LLama model using (Dettmers et al., 2023) QLoRA technique to predict the specificity scores across various metrics. However due to less data in training set, the results of the fine tuned model were worse than the publicly available GPT model.

11 Future Work

Our prompt scoring mechanism has shown promising results, but there are several work for future exploration and improvement. One potential direction is to validate our approach across additional constraints, such as ambiguity and style, to further generalize our prompt score to other domains. Another area of exploration is to expand our evaluation dataset to include hundreds of prompts with increasing specificity, allowing us to calculate more robust average performance metrics for each model. Also, instead of having integral scores, we can have a more powerful evaluation if we try continuous scores for the metrics. This will provide a more comprehensive understanding of the models' capabilities. Additionally, we aim to explore alternative scoring methods, such as utilizing other Large Language Models (LLMs) specifically designed for predicting scores instead of using GPT Eval. This will enable us to compare the effectiveness of different scoring approaches and potentially identify more accurate and efficient methods.

12 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes, we did use ChatGPT to assist in report writing.

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - Write latex code for putting three graphs on a same horizontal line in latex in a 2 column format report
    - ∗ Used to plot the graphs in the Result section
  - Rephrase this paragraph: Next we try to evaluate the following models (Alpaca, Gemma and Claude) on different metrics that we tested (Coherence, Constraints following ability and Fluency). We tested one metric at a time for all three models so as to analyze which model performs better for that particular metric. Following are the three graph for the same
    - ∗ Used to answer in the Result section
  - how to decrease spacing between the list items
    - ∗ Helped in the dataset statistics section to decrease space between list items
  - write a similar prompt to an LLM. $PROMPT$
    - ∗ Used to generate an example prompt based on input prompt
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - Result section
    - ∗ The code provided to put multiple images in a horizontal line didn't work at first place, but on providing the error, it was fixed. Also the paraphrasing required some changes. So overall an average outcome. Mostly used to re-write text.
  - Dataset section
    - ∗ Used to to get latex code and the results were satisfactory.

References

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms. In *International Conference on Learning Representations*. University of Washington. Available at https://arxiv.org/abs/2305.14314.

Kandpal, N., Deng, H., Roberts, A., Wallace, E., and Raffel, C. (2023). Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, pages 1–12.

Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z.-R., Stevens, K., Barhoum, A., Duc, N. M., Stanley, O., Nagyfi, R., et al. (2023). Openassistant conversations - democratizing large language model alignment. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023), Track on Datasets and Benchmarks*. Google Research, Yale University. Available at https://arxiv.org/abs/2304.07327.

Li, X., Yu, P., Zhou, C., Schick, T., Levy, O., Zettlemoyer, L., Weston, J., and Lewis, M. (2024). Self-alignment with instruction backtranslation. In *International Conference on Learning Representations*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. (2023). G-eval: Nlg evaluation using gpt-4 with better human alignment. Microsoft Cognitive Services Research. Available at https://arxiv.org/abs/2303.16634.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Qin, Y., Song, K., Hu, Y., Yao, W., Cho, S., Wang, X., Wu, X., Liu, F., Liu, P., and Yu, D. (2024). Infobench: Evaluating instruction following ability in large language models. Tencent AI Lab, Seattle. Available at https://arxiv.org/abs/2401.03601.

Ravichander, A., Hovy, E., Suleman, K., Trischler, A., and Cheung, J. C. K. (2020). On the systematicity of probing contextualized word representations: The case of hypernymy in bert. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 88–102, Barcelona, Spain (Online). Carnegie Mellon University, Microsoft Research, McGill University.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)*. Google Research, Brain Team.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., and Levy, O. (2023a). Lima: Less is more for alignment. Preprint available at https://arxiv.org/abs/2305.11206.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. (2023b). Instruction-following evaluation for large language models. Yale University, Google. Google Research. Available at https://github.com/google-research/google-research/tree/master/instruction_following_eval.