S&DS 4250 - AI Chatbot Ratings

By: Jake Todd and Sonam Wangchuk

Introduction

Websites such as Chatbot Arena (lmarena.ai) are open source websites where users can converse with two LLMs on any topic they choose, and then vote on which response they prefer between the two models (Model A or Model B). Participants can also choose "Tie" or "Both bad" if they don't have a preference for either model (with “Tie” signifying both models being good and equal, and “Both bad” implying both models were bad and equal, generally). The LLMs are initially anonymous, but are revealed to the user after they vote.

The goal of this case study is to analyze this data to conclude which chatbot(s) appears to perform best in the eyes of the judges and users of this website. This analysis will also help us calculate specific outcome probabilities between two different models, and how these differ based on different categories that the model might be asked to focus on (such as writing, math, code, etc.)

Data

Initially, the data was presented as 7 parquet files that you can find here. Within this dataset, each row was a matchup between two chat bots (e.g. Gemini 2.5 Pro vs Claude Sonnet 4), featuring the conversation, who the user selected as the winner of the matchup, the category of the prompt (creative writing, math, code, etc.) and other auxiliary information. After cleaning the data to be an accessible and usable .csv file, we focused on the following variables:

1. Model A and Model B names
2. Outcome (The name of the winning model, or Tie / Both Bad)
3. A series of binary categorical variables that tell us which skills and topics the conversation involved. Specifically, these were:
    - **is_code** - Whether the conversation involves code
    - **category_tag_math_v0.1_math** - Whether the conversation involves mathematical problems
    - **category_tag_criteria_v0.1_real_world** - Whether the conversation involves real world topics
    - **category_tag_criteria_v0.1_problem_solving** - Whether the conversation involves strong problem solving elements
    - **category_tag_criteria_v0.1_creativity** - Whether the conversation involves some kind of creative thinking (not writing)

- **category_tag_criteria_v0.1_complexity** - Whether the conversation involves complex topics and knowledge
- **category_tag_creative_writing_v0.1_creative_writing** - Whether the conversation involves creative writing

Method

Naïvely, the easiest way to rank models is based on their raw win percentage against other models. Models that win more trivially tend to be better. We can do this by calculating the number of times the model wins divided by the number of match ups that the model is in. In this case, we can treat ties as 0.5 wins, and “both-bad” as not a win. In order to get more detailed information, we can also see the raw win percentages in specific head to head match ups against two models, focusing only on conversations that directly featured those two models of interest.

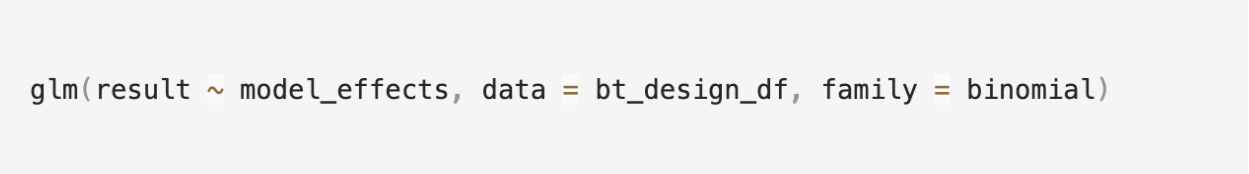
However, this model misses some important information. Specifically, the “quality” of the wins. Here’s an example to make this obvious. Imagine two football players, where Player A has a win rate of 80% and Player B has a win rate of 70%. Player A has been competing against a bunch of middle schoolers, whereas player B is playing in the NFL. According to raw win rates, Player A is a better player than Player B, but this is almost certainly untrue.

In order to account for this, we need to feature a different ranking system that is actively aware of the opponent’s strength as well. Such systems already exist in games such as chess for example, that feature elo systems. We can do something similar here with a Bradley-Terry Model, which gives implicit ability parameters to each model, which thus accounts for opponent strength (beating a stronger model will increase your latent ability parameter more than beating a weaker model, which helps the model implicitly account for the strength of the opponent).

To be specific, the bradley-terry model would have the latent parameters of two models connected to each other with the following formula:

$$P(i \text{ beats } j) = \frac{e^{\theta_i}}{e^{\theta_i} + e^{\theta_j}}$$

This model is estimated using maximum likelihood estimation through a logistic regression model. In R, this was implemented using a code block similar to the following:



```
glm(result ~ model_effects, data = bt_design_df, family = binomial)
```

With this, higher ability scores translate generally to a higher probability of winning against another model.

Finally, we can filter our data to feature specific traits (such as whether we are working with code, mathematics, creative writing, etc.) to see the best models within those categories.

Results

Based on raw win rates, we see the following top 10 models:

Model name	Total Games	Wins	Losses	Win rates
gemini-2.5-pro	9219	6471	2748	0.7019199
gemini-2.5-pro-preview-03-25	1389	932	457	0.6709863
grok-4-0709	1561	1046	515	0.6700833
gemini-2.5-pro-preview-05-06	3255	2097	1158	0.6442396
chatgpt-4o-latest-20250326	7650	4907	2743	0.6414379
o3-2025-04-16	8529	5447	3082	0.6386446
deepseek-r1-0528	6554	4142	2412	0.6319805
grok-3-preview-02-24	6040	3785	2255	0.6266556
gemini-2.5-flash	9668	5966	3702	0.6170873
llama-4-maverick-03-26-experimental	5806	3502	2304	0.6031691

Based on this, we can see that the best model is **Gemini-2.5-pro**, followed by **gemini-2.5-pro-preview-03-25** and **grok-4-0709**. The top 10 models all have win rates of greater than 60%.

Something we can observe here is that different models have a different number of games that they have played, which is a potential reason that the raw win rate might be skewed by the models that you compete against.

Below is a heatmap featuring the head to head win rates for the top 10 models:

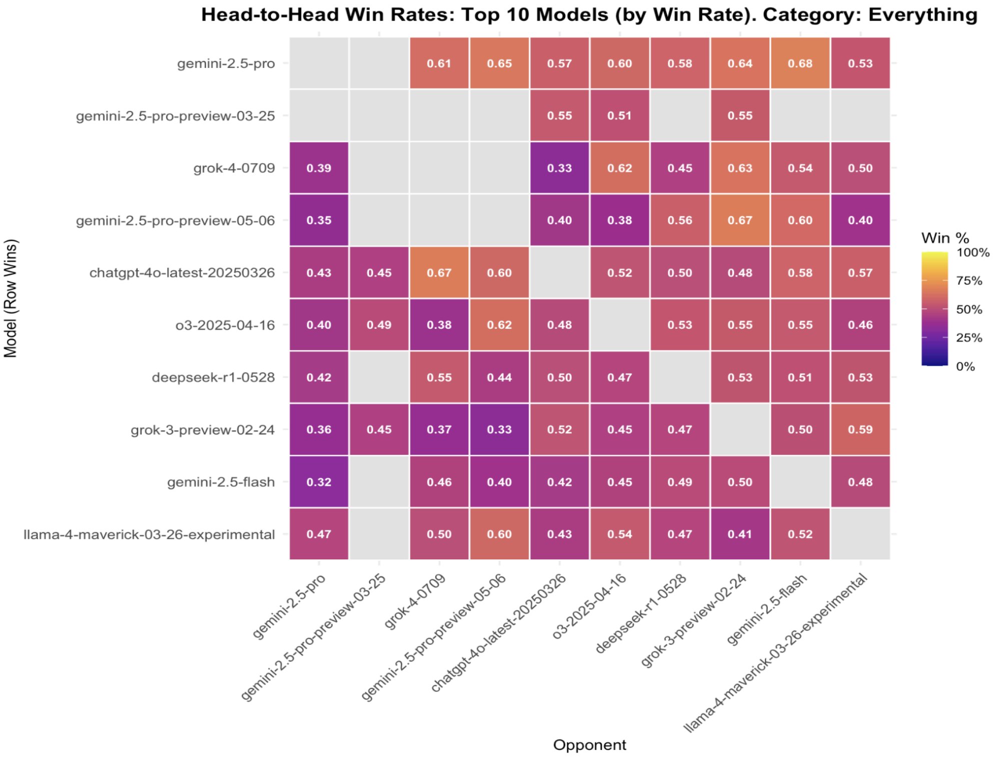


Figure 1: Head to head heat map

Based on the ability scores (for the Bradley-Terry Model), we see the following top 10 models:

Model	Ability
gemini-2.5-pro	0.978255
gemini-2.5-pro-preview-03-25	0.7908489
grok-4-0709	0.7377152
chatgpt-4o-latest-20250326	0.6182392
o3-2025-04-16	0.6145142
gemini-2.5-pro-preview-05-06	0.5928766
deepseek-r1-0528	0.5912983
grok-3-preview-02-24	0.5627368
llama-4-maverick-03-26-experimental	0.5210798
gemini-2.5-flash	0.4948648

There are 2 differences between these two lists:

- **Gemini-2.5-pro-preview-05-06** is ranked 6th on the ability list, instead of being 4th. This means that it is actually more likely to lose against **chatgpt-4o-latest-20250326** and **o3-2025-04-16**, which is something we wouldn't have gotten from the win rate list. It is worth noting that both of these models had significantly more total games than **Gemini-2.5-pro-preview-05-06**.
- **Llama-4-maverick-03-26-experimental** and **gemini-2.5-flash** are flipped on the two lists. In this case, the model with less total games ended up getting a higher ability score than the win rate suggests.

When we begin focusing on specific categories, we can see that specific models are better at specific categories, different from how they generally perform. To see this, we can look at how the ability scores of models when focusing only on *creative writing* problems as an example.

In the barplot below, we can see that **gemini-2.5-pro-preview-03-25**, despite generally having a lower win rate and overall ability score than **gemini-2.5-pro,** performs significantly better on creative writing tasks:

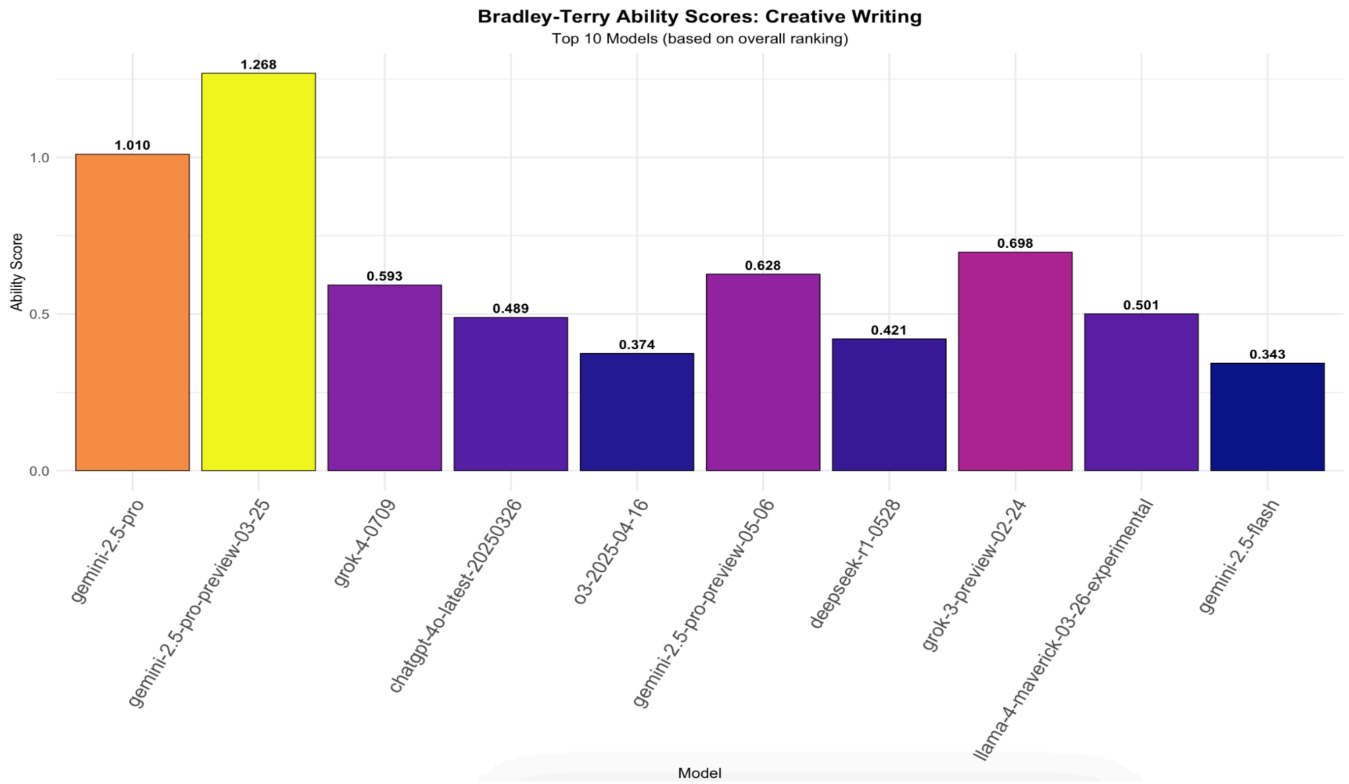


Figure 2: Creative writing BT ability scores

This indicates that different models have different category strengths and weaknesses rather than models generally being better than others. Displaying all of these scores for the different connections in this paper would be tedious and redundant, but they can be found in the following Shiny App (because we are using the free version of shiny apps, this occasionally crashes due to memory issues. There is no way to improve this without paying)

In the Shiny App, there are two primary pages. On the first page, there are dropdown menus to select the topic of the query as well as which type of plot you would like to be displayed. If you select "Bar Plot", then it displays the Bradley-Terry ability scores for the top 10 models in the selected query category. If you select "Heat Map", then it displays a matrix of empirical win percentages for the model (listed in the row) over the opponent model (listed in the column). On the second page (which can be reached by clicking the button at the bottom), one can select two models for a "head-to-head matchup". Once two models are selected (must be different models), a barplot showing the empirical and Bradley-Terry win probabilities is displayed for specifically the matchups of those selected models. One can return to the first page by clicking the button at the bottom.

Discussion

Our analysis captures a reasonable amount of information about these chatbot comparisons. Specifically, we are able to see overall win rates, how these win rates differ based on the strength of the opponents, and see differences based on the category of information and so on.

One of the key insights from looking at the difference between raw win rates and the Bradley Terry model is that we can see how, intentionally or otherwise, certain models might be presented as being better by simply being matched up more against weaker models, and vice versa. Given that, the Bradley Terry model is likely a better way to think about the abilities of these models as it accounts for such a method of "gaming" the results.

Due to excluding "tie" and "both bad" outcomes from the Bradley-Terry model, as these outcomes are not comparative, our empirical win rates and calculated Bradley-Terry win rates (from ability scores) are slightly different. This difference is clearly shown in the "head to head matchups" section of the Shiny App. This is a slight source of inaccuracy, but generally it seems that the empirical win rates and recalculated Bradley-Terry win rates often differ by less than 5%. It seems that situations where the user could not decide which model was better in a matchup does not significantly affect the "relative ability" of the models. However, this is a potential limitation of our model.

One final limitation can be seen by the fact that different models succeed or struggle at varying levels based on the type of prompt that they are presented with. The example of

gemini-2.5-pro-preview-03-25 vs. Gemini-2.5-pro is relevant. The latter model is generally ranked higher than the former, but it's not difficult to imagine that someone might game the system by focusing their prompt categories on creative writing in order to skew the results.

Future studies and analysis would likely need to standardize across different prompt categories in order to ensure that all these skills are evaluated holistically when comparing models, or decide that such holistic comparisons are meaningless and focus on category by category rankings (which to an extent can be explored through the Shiny App above)

Conclusion

The best models, both based on raw win percentage and the Bradley-Terry model can be seen in the tables above. You are also able to view specific head to head match ups using the Shiny App linked to this report. The strength and success of a model depends on the specific category of the prompt, with different models thriving in different categories (such as mathematics, creative writing, etc.).

Factors such as the opponents the model is matched against and the categories that a model is tested on affect their win rates significantly, and may lead to gamification if not checked back against by using methods that consider the relative opponent strength (such as the Bradley-Terry model used in this report) and by being clear and deliberate about the categories tested.