CHATBOT PERFORMANCE AND PROMPT DIFFICULTY

ROBERT BAJORSKI, CATHERINE LEE, AND MIKE PITT

Abstract. We construct a model to predict which of two chatbots will perform better on a given prompt. We also model the difficulty of user-generated prompts on a scale from 1 to 9.

1. Introduction

As the usage of large language model (LLM)-based chat assistants (chatbots) proliferates, it is increasingly important to develop new methods of comparing and evaluating chatbot performance. In this paper, we use data from conversations between 20 different chatbots and humans and user voting on the performance of the chatbots to create a model that predicts which of two chatbots will perform better on a given prompt. This builds on previous work of Zheng et al. [3], who introduce the LMSYS-Chat-1M dataset and explore trends in the performance of 25 chatbots on different prompt clusters. Our findings offer insight into the variance in the capabilities of different chatbots when responding to prompts of varying topics and difficulties.

As researchers are actively developing new benchmarks for evaluating chatbot performance and seeking to tune chatbot performance on difficult prompts, it is useful to understand what factors contribute to the difficulty of a prompt. Therefore, we also model the complexity of user-generated prompts on a scale from 1 to 9. This relates to an extensive body of work studying the effectiveness of different kinds of prompt patterns [2] and analyzing the current capabilities and limitations of chatbot performance on area-specific prompts [1].

2. Description of data

The dataset consists of 25322 cleaned conversations between humans and chatbots. Each sample consists of a question ID, the names of two chatbots, the full conversations between the user and the two chatbots named, the user ID, and the user's vote for the name of the chatbot that performed better on the given prompt. Additionally, there are auxiliary datasets containing 256-dimensional text embeddings for each of the human questions, generated by OpenAI's `text-embedding` model, and evaluated topic models and hardness scores for each question from GPT 3.5. The conversations were cleaned to remove non-English conversations, conversations with multiple rounds, and conversations with toxic or harmful content. Potential sources of bias in the dataset include ambiguity in user voting on the difficulty of a prompt and the winner of a chatbot matchup, and sampling bias from excluding non-English conversations or only surveying users on Chatbot Arena from April to June 2023.

We cleaned our data by imputing missing values. We computed win rate for each model by dividing the number of total wins by the number of total questions faced. GPT-4 had

Date: April 2024.

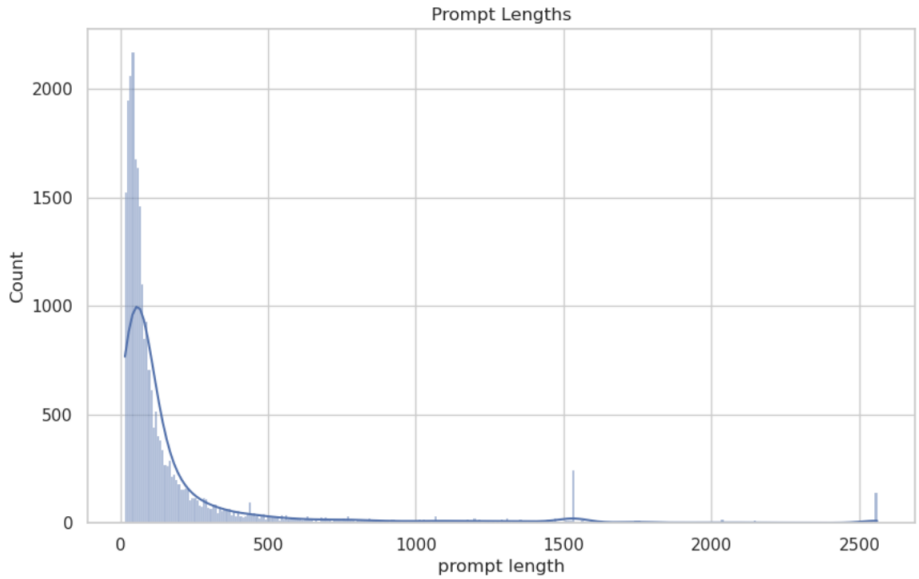

FIGURE 1. Distribution of prompt lengths

the highest win rate of 0.67. We also computed win rate on difficult questions (those with average hardness scores of 9 out of 9). We found high correlation coefficients between win rate on difficult questions and the overall win rate, with a correlation of 0.94. This showed us that models that perform well on hard tasks are also likely to win in general. Moving onto prompt lengths, we found the mean to be 197 however there were large outliers and significant variability, including a maximum length of 2560. For modeling, we removed data in prompt length that were outliers and beyond the 1.5x interquartile range.

We found a similar result for response lengths, with large outliers and a long right hand tail. We created histograms and KDE plots for both prompt length and response length to visualize the distributions. After prompt lengths we computed ELO rating for all of the chatbots. The standard deviation of the ELO ratings was 119 and the maximum was 1233. We found ELO rating to be a good predictor of what chatbot would win a particular engagement. When looking at topic modeling we found the most common topics for the prompts which were problem solving and creativity. When looking at the distribution of battles we found that Vicuna-13b had the most battles between the chatbots and GPT4all-13b-snoozy had the lowest number of battles. We cleaned and processed hardness scores assigned to each prompt. We then averaged multiple scores into a single metric which was analyzed to understand the difficulty of questions and how models fared on harder questions. We also utilized box plots to understand the distributions of response lengths, hardness scores and ELO rating. We created bar plots to display the counts for outcomes, win, lose or tie.

3. Methodology

3.1. **Task A: Predicting the Winning Model.** We used multiclass logistic regression to predict which chatbot model would win the user vote. Given a prompt and two chatbots, model A and model B, we predicted whether whether model A or model B

would win, or whether they would tie (with equally good answers) or tie (with equally bad answers).

We first split the data into training and validation sets using a randomized 80%-20% split. We loaded 256-dimensional text embeddings for each prompt, generated by Open AI's `text-embedding` model. We used PCA to reduce the embeddings to 200 dimensions, capturing approximately 95% of the variance. Then we used K-means clustering to cluster the training prompts. We found that using 25 clusters was optimal for the accuracy rate of the models we later trained on each cluster; using more clusters resulted in overfitting on the training data.

Sampling random prompts from each cluster revealed that our clusters were capturing intuitively meaningful similarities between prompts. For instance, there was a cluster of programming questions, a cluster of creative writing tasks, a cluster of arithmetic questions, a cluster of investing and finance questions, a cluster of basic social interaction tasks, a cluster of historical factual recall questions, and a cluster of philosophical questions, among others. Exploratory data analysis on the clusters also revealed that the clusters differed meaningfully from each other in hardness, the percentage of ties, and the percentage of ties where both models performed poorly. In particular, creative writing tasks and philosophical questions were generally more difficult, and basic social interactions were generally easier. Different models also performed substantially better or worse on certain clusters. For example, although GPT-4 had the highest ELO score overall and an even higher ELO on hard questions, it underperformed on philosophical questions, where it only ranked 6th. We also found that certain clusters were significantly more likely to have ties, or adversarial examples where a model with a lower ELO score performed better than a model with a higher ELO score, and that the percentage of ties (of both kinds) and adversarial examples occurred was similar between the clusters in the training and validation sets. This analysis suggested that it would make sense to train different models on each cluster, so that the importance of each feature could be weighted differently on each cluster, ensuring that our final predictions would be more sensitive to the differences between clusters.

Exploring the data also showed that easier questions were more likely to result in a tie where both models performing well, and harder questions were more likely to result in a tie where both models performed poorly. Therefore we included features related to the difficulty of the question, namely the length of the prompt and the hardness score.

Finally, we included the average response lengths of the two models being compared, because our exploratory data analysis had shown that models with longer responses were more likely to be voted as the winner. We also included the ELO score of each model on prompts with the same hardness score, because we found that there was significant variance in how the models performed across prompts of different difficulties. For example, GPT-4 was ranked first among all the models on questions with hardness scores of 5 and above, and it outperformed the other models more on harder questions. However, on questions with hardness scores of 4 and below, it was ranked behind at least one of Claude-V1 and Claude-Instant-V1.

We trained a logistic regression model to predict the outcome on each cluster. We used the following features to train each model:

(1) Length of the prompt

(2) Hardness score of the prompt (imputed to be the mean hardness score of 7 if missing)
(3) ELO score of model A on the entire training set
(4) ELO score of model B on the entire training set
(5) ELO score of model A on the cluster (if model A responded to fewer than 10 prompts in the cluster, we set this to be the ELO score on the entire training set)
(6) ELO score of model B on the cluster (if model B responded to fewer than 10 prompts in the cluster, we set this to be the ELO score on the entire training set)
(7) ELO score of model A on all prompts with the same rounded hardness score
(8) ELO score of model B on all prompts with the same rounded hardness score
(9) Average response length of model A
(10) Average response length of model B

To predict the outcome of a particular battle, we first projected the textual embedding of the prompt onto the principal components we computed from our training set, then assigned the reduced embedding to one of our predetermined clusters of embeddings, and then used the logistic regression model trained on that cluster to output a prediction.

3.2. **Task B: Predicting the Hardness Score.** For predicting hardness scores, we chose to use a linear regression model. Linear regression is very interpretable and accurately describes the relationships between our features and the hardness score. We rounded the continuous outputs of the linear regression model to generate integer hardness scores from 1 to 9.

For feature engineering we created numerous features that captured the underlying trends of the data. The features we created include:

(1) Prompt length (number of words)
(2) Prompt length (number of characters)
(3) Sentiment score
(4) Subjectivity score
(5) Flesch reading ease
(6) 20 clusters of embeddings
(7) Topic modeling (topics with at least 2 occurrences)

We started with some simple features like counting the number of words and characters in the prompts. Both features were correlated to the hardness score and as such would be good predictors. We also added in sentiment and subjectivity scores for the prompts. The sentiment feature calculates the positivity or negativity of the prompts. Sentiment, and especially extreme sentiment like highly negative or highly positive prompts come with extra complexity. Because of their emotional complexity, the prompts with high negative or high positive scores may be difficult for the chatbots to understand, this is why we included sentiment in the features. Prompts with high subjectivity (e.g. prompts including many user opinions) are more difficult for chatbots to understand, whereas more factual or objective questions are more easily understood by chatbots. Because subjectivity creates complexity in questions we also included this as a feature. We also calculated a complexity score of the prompts based on the Flesch reading ease. The score is calculated based on a combination of length, sentences and syllables. As such this feature directly describes the complexity of the prompts. We also incorporated the embeddings by creating 35 clusters of embeddings using K-means clustering. Embeddings can capture

subtleties in the prompts including semantic meaning, complexity and simplicity. For example embeddings can capture idioms which can be difficult for chatbots to understand. We then one-hot encoded these clusters into our dataset. Finally we included the topic modeling dataset in our analyses. Topic modeling is a proxy for complexity, for example common sense reasoning, expert knowledge and emotional analysis are usually the most difficult for chatbots to answer. By including topics that appeared at least twice, we were able to capture large swaths of this information in the model.

To improve model performance, we standardized the numeric features in our data frame. The standardization scaled the features to have a mean of zero and a standard deviation of one, as our data has varying magnitudes, units and range. Standardization can help with multi-collinearity. Some of our features like word count, prompt length and Flesch reading ease are closely related, as such standardization helps reduce the chances of multicollinearity. Standardization also allows for easier interpretation of the model as all the scales are the same.

4. Summary of results

4.1. **Task A: Predicting the Winning Model.** After standardizing our features for greater interpretability, examining the coefficients of the model on each cluster showed that the prompt length and score were more important for determining the tied classes than for the untied classes. In general, model ELO, model ELO on the cluster, and model ELO on prompts with the same hardness score were the most important features, but the respective coefficients of these features differed significantly from cluster to cluster. The interpretability of the differences in coefficients is somewhat limited by the fact that there is a high level of collinearity between these features; their pairwise correlations are over 0.9. However, our model outperformed the baseline models that predicted the winner based on any of these three pairs of features individually, suggesting that it is capturing some genuine variance between clusters as to which features are the most important.

We experimented with using other models, including a random forest classifier model; a logistic regression model that was trained on all the data (not segmented by cluster) with additional features such as the percentages of ties, bad ties, and adversarial examples (where the model with lower ELO won) in each cluster; logistic regression models trained on each cluster that ensembled the outcomes of simpler baseline models (described in section 5); and an ensemble of two binary logistic regression models, which we describe in further detail below. Our model outperformed each of these models on the validation data. The random forest classifier significantly overfitted on the training data, while the other models produced similar but slightly lower accuracy scores on both the training and validation data.

The ensemble of two binary logistic regression models was the most interesting alternative we considered. The first of the two models was trained to predict the winner on data without ties, outputting either model A or model B. The second model was trained only on tied battles, to distinguish between good ties and bad ties. We used the predicted probabilities to manually tune indeterminacy thresholds centered around 0.5 for the first model, where we would predict a tied outcome and use the second model to predict the kind of tie. However, this model was ultimately unsuccessful because the distribution of predicted probabilities on the tied data was too similar to the distribution of predicted probabilities on the untied data. Because there were more than twice as many untied

battles than tied battles, this meant that predicting any ties at all decreased the accuracy score substantially, because any indeterminacy threshold would capture more untied battles than tied battles.

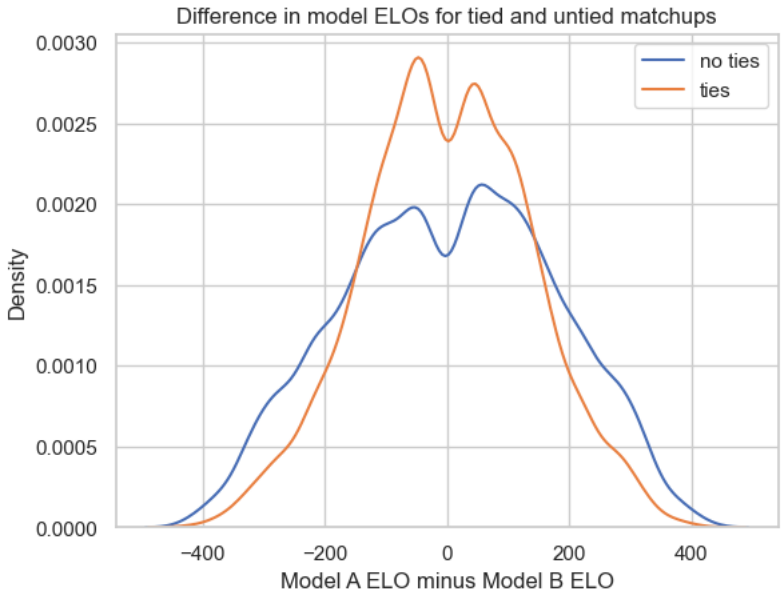

FIGURE 2. Distribution of differences in ELO

As a simplified example of this phenomenon, in figure 4.1 we show the KDE plot of the difference in ELOs between model A and model B on the tied data and the untied data respectively. The KDE curve of the tied data has a slightly higher and narrower peak than that of the untied data, indicating that, as we might intuitively expect, models with similar ELO scores are slightly more likely to tie against each other than models with very different ELO scores. However, there is not enough separation in the KDE curves to define a threshold where it is more likely for two models to tie than for one of them to win, especially because it is overall more than twice as likely for a battle to result in a winner than a tie.

4.2. **Task B: Predicting the Hardness Score.** After defining the features and transforming these where it made sense, we explored different regression models to accomplish the lowest loss possible. We started with a linear regression model as our baseline, which had a MSE of 3.5. Linear regression is a highly interpretable model that would give us insight into how our features performed. For example, we could look at the coefficients and easily understand what the most and least important features were. This was made even easier by standardizing all the numeric features to compare importance. The feature importance scores are below. It is interesting that the most important features are associated with length of the prompt and response, these are things we included in our base model and that we found in our early EDA were important for predicting hardness score.

We also explored using logistic regression models to predict hardness scores. We tried two approaches: a multiclass logistic regression model that predicted a binned hardness score (low, medium, or high); and a threshold approach using an ensemble of 9 binary

classifiers, one for each numerical score, that predicted whether a given prompt's hardness score was above or below the given number. We achieved a ROC of 0.70 in the bin logistic approach with an accuracy score of 0.60. However, it was a nontrivial task to transform the prediction of the multiclass regression into a numerical score. We tried a majority vote approach to calculate an expected value of each class prediction. We would assign a median value for each class and multiply that against the probabilities of the multiclass model. This ultimately did not perform as well as the linear regression model. The threshold approach was successful at predicting a binary classification for each numerical score and accuracy scores were high for each model, but when combined into a single model, the model had higher MSE than linear regression.

Feature	Importance
log_prompt_length	1.023984
square_sentiment	-0.842829
log_word_count	-0.409146
subjectivity	0.318256
topic_cluster_5	0.282826
topic_cluster_2	0.207631
topic_cluster_16	-0.188652
topic_cluster_11	0.131599
topic_cluster_1	0.131032
topic_cluster_19	0.099715
topic_cluster_4	0.095947
topic_cluster_12	0.076605
topic_cluster_3	0.075392
topic_cluster_8	0.075283
topic_cluster_6	0.064288
topic_cluster_18	-0.057234
topic_cluster_9	0.030806
topic_cluster_7	-0.022664
topic_cluster_14	-0.016247
topic_cluster	0.007219
topic_cluster_10	-0.006448
topic_cluster_15	0.006013
topic_cluster_17	0.004064
topic_cluster_13	0.003021
flesch_reading_ease	-0.000887

TABLE 1. Baseline model feature importance scores

5. DISCUSSION

5.1. **Task A: Predicting the Winning Model.** We compared our model to the following three baseline models:

(1) Predict that the model with higher ELO wins
(2) Predict that the model with higher ELO on prompts in the same cluster wins
(3) Predict that the model with higher ELO on prompts with the same score wins

The accuracy scores of the models on the training and validation sets are shown in Table 5.2.

	Training	Validation
Our model	0.565	0.544
Model (1)	0.530	0.528
Model (2)	0.541	0.530
Model (3)	0.533	0.525

TABLE 2. Baseline model accuracy scores

From Table 5.2, we can see that our model outperformed each of the baseline models on both the training and validation sets. Our model had a slightly higher accuracy score on the training set than on the validation set, but this difference was not large enough to suggest that overfitting was an issue.

The precision and recall scores for our model on each of the four outcome classes is shown in Table 5.1.

	Model A	Model B	Tie	Tie (both bad)
Training precision	0.573	0.575	0.429	0.495
Training recall	0.735	0.717	0.036	0.258
Validation precision	0.553	0.552	0.381	0.476
Validation recall	0.714	0.703	0.029	0.242

TABLE 3. Model precision and recall

The data in Table 5.1 show that our model was substantially better at identifying the winning model in battles that did not result in a tie. In particular, our model had very low recall ($\sim$3%) on tied battles where both models performed equally well.

Examining the confusion matrices shows that battles where model A won were most commonly misclassified as wins for model B, and vice versa. When we trained our model only on data not containing ties, we achieved accuracy scores of about 77% and 75% on the training and validation sets respectively. This shows that, even setting aside ties, our model is not completely able to distinguish battles where model A won from battles where model B won.

One interesting finding from the confusion matrices is that, despite the low recall for the tied classes, our model very rarely misclassified bad ties as good ties. This shows that our model is better at distinguishing bad ties from good ties than it is at distinguishing tied battles from battles with a winner.

From a societal perspective, when we think of the best chatbots, or the most winning chatbots, its important to understand that the best chatbots may be fostering a dependence on them because they are so good and because they win so much. For example, many people now use ChatGPT for everyday tasks, but there are potential downsides. Its also important to note some potential concerns for the modeling where bias might be introduced. For example, users who rate a chatbot as winning may then be more likely to rate that same chatbot as thew winner in later conversations, introducing bias. Ethically its possible that some people, especially those with lack of Internet access, lack of ability to pay for chatbots or for some other misfortune, cannot interact with chatbots

and therefore cannot receive their benefits. There are also concerns around user privacy with chatbots, and how the data that is being fed into them by users, is being used for retraining the chatbot itself. Are there conversations that the chatbot should not be using for training that comes from users? Even if the user agrees to sending the data in? Perhaps the user has no other avenue to get advice or talk about personal aspects of their life with anyone else, a chatbot provides an unbiased adjudicator. These data privacy concerns are among the top ethical issues facing chatbot creators today.

5.2. **Task B: Predicting the Hardness Score.** We used a linear regression model to predict the hardness score, because the linear regression model we trained was ultimately more successful than the logistic regression models we attempted. However, linear regression assumes that changes in the predictor leads to consistent changes in the dependent variable, which might not hold true because our dependent variable is ordinal. The relationship between topic variables and an ordinal score might not necessarily be linear. Logistic regression does not assume a linear relationship between the dependent and independent variables, making it more flexible for categorical and ordinal data. In the future it may be advisable to attempt to train a logistic regression to achieve this. Our team also did not try other modeling methods like XGBoost, decision trees, or classification, the performance of the model may benefit from these different modeling approaches. Finally, we were computationally constrained, as such, our model was not optimized based on the features we had at hand. We worked on reducing the dimensional of these features, however, we believe that the model would have performed better had we been able to use all the topic modeling classes and all the embeddings, instead of clustering them. In the future, to improve performance, these methods may be considered. From a societal perspective, as chatbots are able to answer more and more complex questions it may lead to a loss of interpersonal communication. Especially in the cases where a chatbot can answer questions akin to a complex field like therapy. Refining the performance of chatbots when it comes to complex questions with high hardness with further this. Its also important to note that there could have been some biases in the training data we are unaware of. These biases can affect the outcome and feasibility of the model. for example a model that has a very biased training set in some way shape or form, will not generalize well and could lead us as researchers into poor conclusions. These impacts should be considered when viewing this model.

	Training	Validation
MSE	2.37	1.53
RSME	1.95	1.39

TABLE 4. Linear regression loss

REFERENCES

[1] Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017, September 2023.

[2] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt, 2023.
[3] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset, 2024.