Predicting Human Preferences for LLM Response Enhancement

Zehra Marziya Cengiz, Erik Papernuik, Elias Saker (**Team 13**)

GitHub

1. Introduction

With LLMs becoming widely used, understanding why humans prefer one response over another has become an important question. This report addresses that question through the Kaggle LLM Classification Finetuning competition, where the task is to predict which of two responses users prefer. We describe our modeling process, key experiments, and results.

2. Dataset and Task Description

The dataset from the Kaggle LLM Classification Finetuning competition contains 57,477 training samples. Each sample includes a user prompt, two different LLM responses from 33 different models, and human preference data. The data reveals that Model A wins (34.91%), Model B wins (34.19%), or it's a Tie (30.90%). This means the classes are reasonably balanced, showing that the dataset is fairly randomized and unbiased.

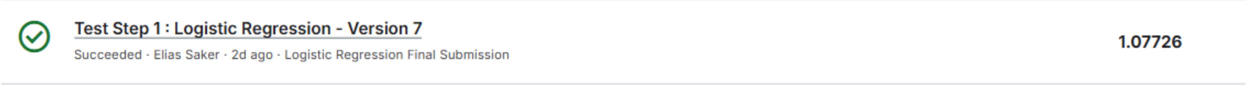
For evaluation, the competition uses Log Loss, which penalizes confidently incorrect predictions more harshly. Additionally, the data revealed that response length has a positive correlation with win rate ($r = 0.499$, $p < 0.001$) for some models. Lexical and structural features also influence human preference. For example, paragraph count ($r = 0.143$), list formatting ($r = 0.072$), and quote usage ($r = 0.062$). In conclusion, the findings show that human preference is affected by both content quality and presentation clarity.

3. Baseline Methods

- **3.1 Lexical/Length Features Baseline**

From our initial Exploratory Data Analysis, we identified a small set of structural differences between responses, such as length and formatting patterns, that showed consistent, though moderate, association with user preference. Using these features, we trained a multinomial Logistic Regression model to classify outcomes (A wins, B wins, or Tie).

The baseline achieved an average log loss of ~1.07, representing a solid starting point that confirms these simple structural cues carry meaningful predictive information. This provides a clear and well-founded reference level from which we now progress to richer embedding-based models.



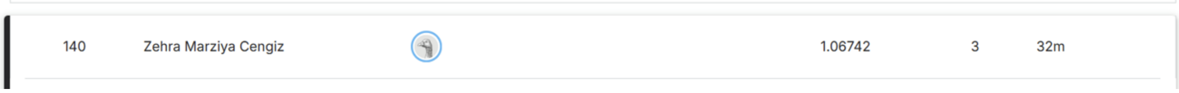
Test Step 1 : Logistic Regression - Version 7

Succeeded · Elias Saker · 2d ago · Logistic Regression Final Submission

1.07726

- **3.2 Embedding-based Baseline**

To capture semantic relevance, we used the pretrained all-MiniLM-L6-v2 model to embed prompt–response pairs. We concatenated embeddings for Response A and B and trained a Logistic Regression classifier. This model achieved ~1.072 log loss. The improvement was modest, suggesting that semantic and structural cues are complementary, motivating combined modeling.The embedding-based model had a mean log loss of 1.072 (±0.002). This result showed us that lexical features successfully captured important human preference trends that the embedding model was not catching, such as response length.



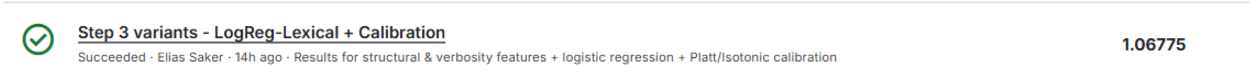
140 Zehra Marziya Cengiz 1.06742 3 32m

4. Model Extensions and Key Features

In the previous section, we established two initial baselines: a lexical/length feature model and an embedding-based semantic classifier. While these models provided meaningful insights, they lacked the ability to directly compare prompt–response interactions at a deeper language level. Therefore, Step 3 focused on extending the modeling pipeline along three directions: **feature engineering**, **fine-tuned transformer models using LoRA**, **ensembling and calibration**.

- **4.1 Feature Engineering**

After establishing the baseline, we extended the feature set to account for even more known biases in human preference. In particular, we included verbosity features (response length, word count, paragraph count) and position bias indicators (whether the preferred response tends to appear in position A or B). These features were motivated by initial correlations showing that longer, more structured responses often received higher preference. We integrated these signals into the logistic regression model and observed a small but consistent improvement in log loss. However, the effect size remained limited, suggesting that stylistic and formatting cues alone cannot fully explain user preference and that deeper semantic understanding is required.



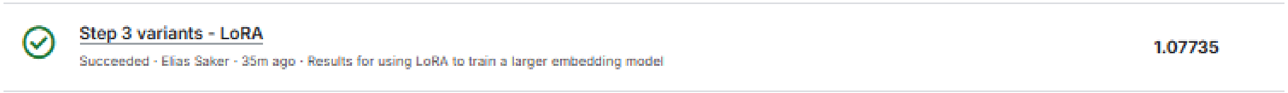
Step 3 variants - LogReg-Lexical + Calibration

Succeeded · Elias Saker · 14h ago · Results for structural & verbosity features + logistic regression + Platt/Isotonic calibration

1.06775

- **4.2 Embedding Model (LoRA)**

In addition to our previous language model, we also experimented with LoRA fine-tuning on a larger pretrained transformer to allow the model to learn preference patterns directly. LoRA adds a small number of trainable parameters while keeping most weights frozen, which is ideal under GPU limits. However, we observed that the LoRA model showed limited improvement and performed below the embedding-based classifier. This suggests that meaningful gains from LoRA would likely require more compute or longer training than available in this setting.



Step 3 variants - LoRA

Succeeded · Elias Saker · 35m ago · Results for using LoRA to train a larger embedding model

1.07735

- **4.3 Ensemble and Calibration**

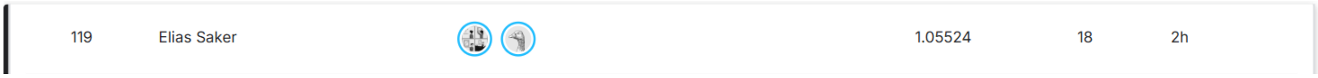
Since different models captured different aspects of the task, we tried combining them in a weighted ensemble. The lexical + bias-aware model contributed structural information, while the embedding model contributed semantic understanding. We then applied temperature scaling to calibrate output probabilities, reducing overconfidence and improving log loss. The ensemble achieved the best performance in our validation experiments, outperforming all individual models. This confirmed that human preference is influenced by multiple complementary factors, and no single modeling approach was sufficient on its own. Unfortunately, technical issues prevented us from submitting the predictions made by this approach on Kaggle.

5. Results

For classical models (lexical and embeddings), we used 5-fold stratified cross-validation with calibrated probabilities (OOF). For the LoRA model, we used a stratified 85/15 train–validation split due to training constraints. The ensemble combined CV-OOF predictions with LoRA's pseudo-OOF and was calibrated via temperature scaling.

Model	Mean Validation Log Loss	Kaggle Log Loss
Lexical / Length Baseline	1.07133	1.07726
Calibrated Embedding + Classifier	1.05812	**1.05524**
Ensemble	**1.049**	Unable to compute

Our best competition performance was achieved using the pretrained embedding-based model of step 2 with isotonic calibration and adding a simple classifier on top of the concatenated embeddings of Response A and Response B to predict which response would be preferred.



119 Elias Saker 1.05524 18 2h

In terms of validation, the ensemble model, that combined the strengths of both the lexical+bias-aware model and the calibrated embedding-based model, was the most performant.

6. Error and Bias Analysis

- **6.1 Quantitative Analysis**

We evaluated 3 different models on the validation set: Lexical (Isotonic), Embeddings (Isotonic), and Ensemble (Weighted + Temperature Scaling) from step 3. The ensemble model achieved the best performance with a log loss of 1.049 and accuracy of 45.6%. Unfortunately, the improvement was minimal (~1.2%), showing that for significant improvement we need a different approach.

We analyzed per-class log loss which showed us that the "Tie" class was the hardest to predict, meaning the model struggled with ambiguous cases where responses are similarly good. Interestingly, the ensemble model, which is our best model, had slightly worse performance on the Tie class.

Another analysis method we used was Confusion Matrix Analysis, which helped us see error patterns for each model. The embeddings model predicted ties more often and had 24% accuracy on the Tie class. The ensemble model still struggled with the Tie class, correctly identifying only 18% of true ties. This again confirmed that the weakest part of the models is predicting the Tie class.

The main conclusion from the confidence analysis is that our models weren't confident about their predictions. This can be caused by calibration issues or real uncertainty in most cases.

- **6.2 Qualitative Analysis**

In this part, we examined examples of misclassified cases. One clear finding was that when all models failed, it was about ambiguous comparisons where responses had similar length, quality, and style. For instance, when asked about handling a late restaurant order, both responses provided reasonable answers. However, all models chose "Model B Wins" when it was actually a tie.

We analyzed position bias and verbosity bias. Position bias revealed that the models want to declare a winner even when the responses are tied and both good quality. The verbosity bias proved something we were suspecting from the start, which is that the model thinks the longer the answer, the better the quality of the response.

7. Reproducibility Notes

Development and testing of the model was made in 2 different environments. 1. Local with VS Code using Python 3.10.11. 2. Kaggle Notebooks using Python 3.11. The dependencies are specified in requirements.txt. Since we used different environments, GPU/CPU usage depended on which environment was used. Runtime took approximately 30 min for each model in Kaggle. In the local environment it was 5–10 min.

The biggest issue we faced was the environment used in the Kaggle competition because there was no-internet constraint. This created different types of errors. One of them was that we weren't able to use download pre-trained models from HuggingFace. As a solution, we uploaded the sentence-transformers/all-MiniLM-L6-v2 model as a Kaggle Dataset and loaded it locally within notebooks.

8. Limitations and Future Directions

Our last model achieved only ~1.2% improvement over baseline and had poor performance on the Tie class. There were biases toward longer responses, which suggested the model captured presentation over true quality. In the future, larger model fine-tuning could improve the model's performance.

9. Conclusion

Predicting human preference is a challenging task, as can be seen from this report. Even though different methods were used from structural features to semantic content, we weren't able to improve our results significantly. Our best competition result is with the embedding model (log loss 1.05524), and best validation: performance is with ensemble model (log loss 1.049, 45.6% accuracy). These results showed us that for a substantial improvement on the model, we will need to try different methods.