Machine Learning Project
Project #2

20211109 배현우
20213948 최원용
20203901 지승후

1. Description of Dataset and Task

The dataset consists of train.csv and test.csv. train.csv includes the following columns: id, model_a/b, prompt, response_a/b, and winner_model_a/b/tie. Here, prompt represents the input given to the LLM, model_a/b indicate the names of the LLMs used, and response_a/b are their respective outputs. winner_model_a/b/tie specifies which model's response was preferred.
test.csv contains id, prompt, and response_a/b.
The task is to predict the preference label (a, b, or tie) for each example, given the prompt and the corresponding responses from both models.

2. Baseline model and Model extension

The baseline model is Logistic Regression, and the features are derived from the lexical and length characteristics of response_a and response_b. Specifically, we compute the character length, word count, token count, and average word length, and use the difference between the values of model A's and model B's responses as features.
In addition, MiniLM is used to convert each prompt + response_a/b pair into embedding vectors, and the difference between the two embeddings is also included as part of the feature set.

3. Key features, Embeddings, or Model used

We leverage DeBERTa-v3-small fine-tuned using LoRA for a 3-way classification task. The input, consisting of prompt + response_a + response_b, is tokenized and encoded into contextualized embeddings via the DeBERTa model. The additional features include character-level length, word-level length, sentiment, and sentence style (number of exclamation marks, question marks, and commas). Similar to logistic regression, we utilize the differences between response A and response B for these features.

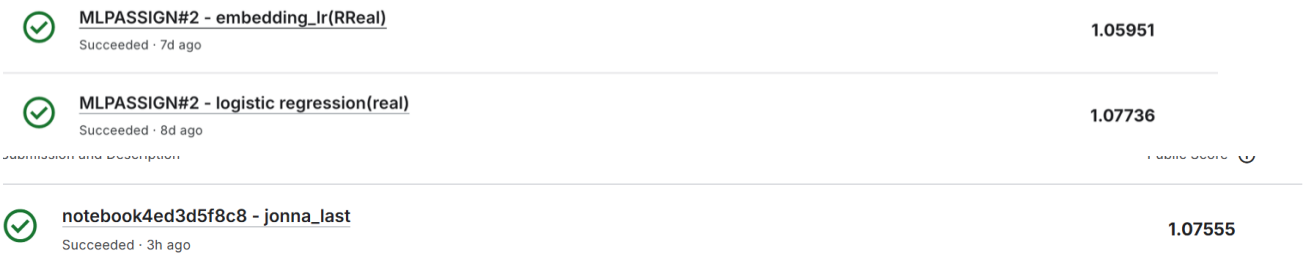
4. Validation strategy and Results

The dataset was split into train (80%) and validation (20%) sets using stratified sampling by label. Validation performance was evaluated using the log loss metric across 5 training epochs. The model combined DeBERTa-v3-small + LoRA embeddings with bias-aware features (e.g., sentiment, length, and punctuation differences). The best model was selected based on the lowest validation log loss achieved during training.
The model achieved a best validation log loss of 1.0419 after epoch 3. Compared to the baseline logistic regression model 1.07736, the DeBERTa+LoRA hybrid reduced validation log loss by 3.4%.

5. Performance comparison table

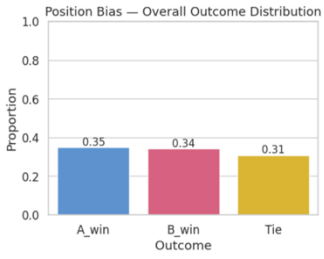
Model	Performance(log loss)	Base Encoder	Input
Logistic Regression (baseline)	1.07736		Lexical/length features
Logistic Regression with embedding	1.05951	MiniLM-L6-v2	Prompt + Responses (embeddings)
deBERTa + LoRA	1.0419	DeBERTa-v3-small	Response pairs with bias-aware features

6. Kaggle leaderboard score (screenshot)

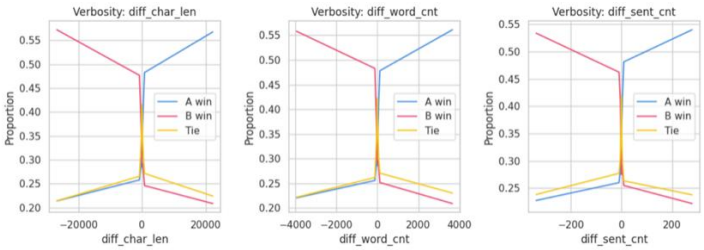


7. Error and bias analysis

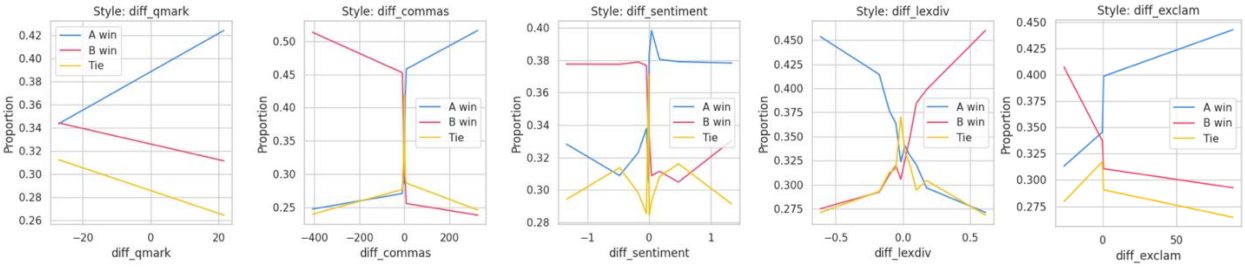
Position bias refers to the tendency for the selection probability to vary depending on the position (left or right) where each model's response is displayed. Assuming that 'model a' is shown on the left and 'model b' on the right, the difference in their win rates is 0.01, which is not statistically significant.



Verbosity bias refers to the tendency for the selection probability to vary depending on the length of each model's response. In our project, we analyzed character-level length, word-level length, and sentence-level length. Across all three measures, we observed that as verbosity increases, the likelihood of a model being selected also increases.



Style bias was analyzed based on the number of exclamation marks, question marks, and periods in each model's response, as well as sentiment and lexical diversity. For sentiment analysis, we used **VADER**, a sentiment lexicon–based analyzer, and found that a higher sentiment score corresponded to a higher selection probability. Similarly, greater lexical diversity was also associated with a higher likelihood of being selected. Regarding punctuation, both exclamation marks and periods showed a positive correlation with selection probability, whereas question marks were associated with consistently higher win rates for model A, regardless of their count.



8. Reproducibility notes (runtime, environment, seeds, handling no internet constraint)

The experiments were conducted in the "GPU T4 x2" environment, one of the Kaggle accelerator settings. To ensure that the project could run in an offline environment without internet access, the MiniLM, VADER, and DeBERTa-v3-small models were downloaded locally and then uploaded to the Kaggle input directory for use.

9. Limitation and Possible future directions

Although relatively lightweight models such as MiniLM and DeBERTa-small were used, performance could be improved by utilizing models with a larger number of parameters.
It is also possible to predict user preference by analyzing the prompt to determine how well the response reflects the information requested by the user.

10. Github URL

https://github.com/BHW-1224/MLP_Proj2_Team2.git