Team Project 2

Finetune LLMs to Predict Human Preference

20203350 Soogon Kim
20202437 Sunwoo Bang
20202087 Minseok Yoon
20191807 Seokhun Jung

Github: https://github.com/CSE-MLP/TeamProject2

1. Baseline Model(TF-IDF + Linear Regression)

	LATEST SCORE	BEST SCORE
LLM Classification Finetuning	1.17044 V2	1.17044 V2

We first tested the linear regression model as a baseline. Text data were embedded using TF-IDF vectorization to capture both lexical and stylistic features of each response.

Multiclass logistic regression model predicts the probability of model A, B or a tie being preferred. The final Kaggle submission achieved a loss score of 1.17044.

2. Embedding-based Model

two input DeBERTa-v3-small + Logistic Regression

The dataset was constructed as follows for the training.

<CLS> prompt <SEP> response_a <EOS>
<CLS> prompt <SEP> response_b <EOS>

The processing procedure of the model is as follows.

1) prompt + res_a → DeBERTa → CLS1 (768-D)
2) prompt + res_b → DeBERTa → CLS2 (768-D)
3) [CLS1 | CLS2] (1536-D) → Logistic Regression → 3 classes

Results

Training Validation Log Loss: 1.03408

Submission

Max input length was changed 512 -> 256 of embedding for Kaggle submission to prevent Out-Of-Memory problem.

Competition Notebook	Public Score	Best Score
LLM Classification Finetuning	1.05221	1.05221 V4

3. Model Extensions

one input DeBERTa-v3-xsmall classification

The dataset was constructed as follows for the training.

<CLS> prompt <SEP> response_a <SEP> response_b <SEP>

The processing procedure of the model is as follows.

[<C>, prompt, <S>, res_a, <S>, res_b, <S>] → tokenizer → input_ids (batch_size, 510)
→ DeBERTa → CLS (batchsize, 1, 768) → classifier → logits(batchsize, 3)

one input DeBERTa-v3-xsmall classification fineturning

The fine-tuning configuration is as follows.

epochs	3	batch_size	256
max_len	510 = 1 + 169 + 1 + 169 + 1 + 169 + 1	lr	5e-5,

The results of the fine-tuning are as follows.

lr	train loss	train acc	valid loss	valid acc
5e-5	0.9508	0.4964	1.0416	0.4792

- The model showed a relatively low maximum accuracy of 0.479.
- When the *prompt*, *res_a*, and *res_b* were all combined into a single sentence, it is likely that the content was not fully reflected due to the maximum input token length being limited to 512 tokens.

Data analysis

	prompt_tokens	response_a_tokens	response_b_tokens	total_tokens
count	57477.000000	57477.000000	57477.000000	57477.000000
mean	102.043948	345.802547	348.628965	796.475460
std	320.134878	419.311586	433.176993	912.143676
min	7.000000	5.000000	5.000000	20.000000
25%	17.000000	100.000000	100.000000	307.000000
50%	27.000000	264.000000	266.000000	601.000000
75%	64.000000	450.000000	452.000000	960.000000
max	9359.000000	21895.000000	17693.000000	39294.000000

- More than 50% of *res_a* and *res_b* contain over 260 tokens.
- It was assumed that using the existing approach could lead to potential issues when analyzing the content of more than half of the responses.

two input DeBERTa-v3-xsmall classification

To incorporate a greater number of response tokens, a model was designed to process the data by dividing it into two parts.
The dataset is as follows.

<CLS> prompt <SEP> response_a <SEP>

<CLS> prompt <SEP> response_b <SEP>

The processing procedure of the model is as follows.

1) [<C>, prompt, <S>, res_a, <S>] → tokenizer → input_ids (batch_size, 510)
→ DeBERTa → CLS1 (batchsize, 1, 768)

2) [<C>, prompt, <S>, res_b, <S>] → tokenizer → input_ids (batch_size, 510)
→ DeBERTa → CLS2 (batchsize, 1, 768)

3) concat [CLS1, CLS2] → CLS (batchsize, 1536) → classifier → logits(batchsize, 3)

two input DeBERTa-v3 classification fineturning

The fine-tuning configuration is as follows.

epochs	3	batch_size	256
max_len	512 = 1 + 69 + 1 + 440 + 1	lr	5e-5

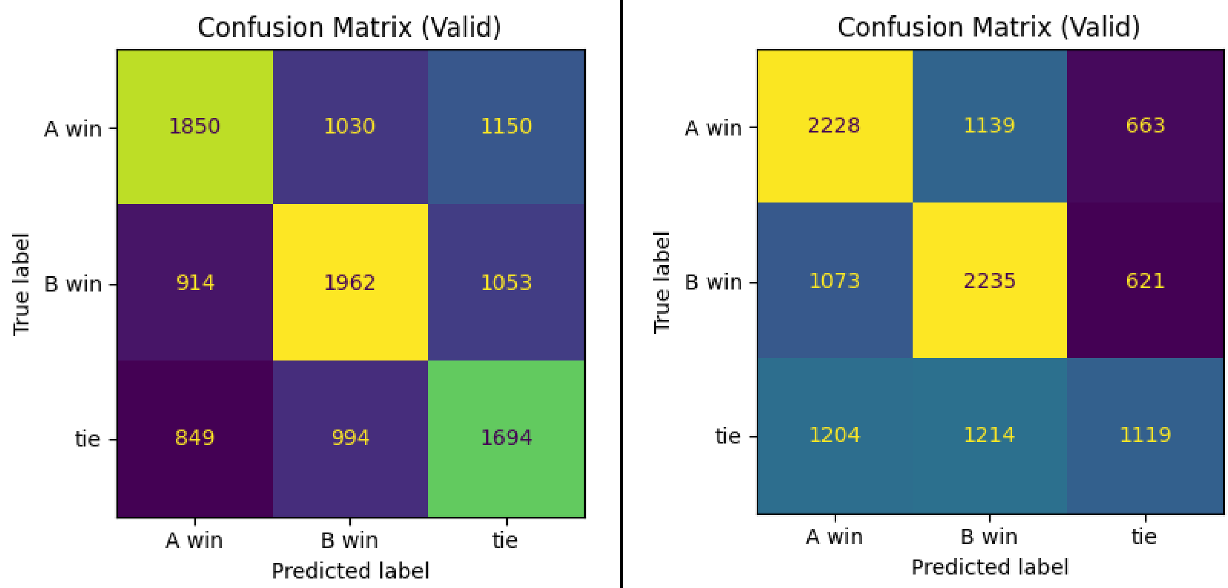
The results of the fine-tuning are as follows.

lr	train loss	train acc	valid loss	valid acc
5e-5	1.0530	0.5495	1.0360	0.4855

- The maximum accuracy improved to 0.485. but, the overall performance remains relatively low.
- This suggests that rather than verifying all responses, increasing the model's capacity or scaling up its architecture would likely be a more effective approach to enhance its analytical capability.

4. Error Analysis

Initially, a single-input approach was tested, concatenating prompt and both responses into one sequence.
However, due to the 512-token limit, more than half of the samples suffered from truncation.
Therefore, we adopted a two-input architecture that processes each response separately and combines the CLS representations for comparison.



one input deberta-v3 classification model **two input deberta-v3 classification model**

- The **one-input model** demonstrates relatively balanced performance across the three prediction categories: *A win*, *B win*, and *tie*.
- The **two-input model** exhibits superior performance in predicting *A win* and *B win*, but its performance in predicting *tie* is comparatively weaker.
- This suggests that the two-input model's ability to analyze a larger number of tokens enables it to more accurately identify cases that were previously classified as *tie*.
- However, due to its reduced capability to jointly analyze *prompt*, *res_a*, and *res_b*, it can be inferred that the model experienced a loss in its ability to accurately evaluate *tie* cases.

5. Final Model and Results

We submitted the fine-tuned two-input DeBERTa-v3-xsmall classification model to Kaggle as our final submission, and the results are as follows.