Yijun Liu

LLMs-Based Persuasive Argument Prediction

Methods

1. Dataset

   In this experiment, I used a subset of the original full dataset from the pair_task folder due to cost restraints. I randomly sampled 15% of the training and testing sets, resulting in 518 data points in the training set and 121 data points in the holdout set. The datasets are combined, resulting in a new dataset containing 639 unique discussions.

2. Problem Definition

   This study aims to predict which rooted path-unit changed the original poster's mind from two rooted path-units (a pair) for each discussion. A rooted path unit is defined as all replies in a path by the root challenger, where the root challenger is the author of a direct reply to the original post. The false rooted path-unit is another rooted path-unit in the same discussion tree as its pair true rooted path-unit. It has the highest Jaccard similarity compared to its true rooted path-unit, determined after removing stopwords from each reply.

   In a rooted path-unit, there is the root reply (the rote challenger's direct reply to the OP) and other replies under the root reply. In the following section, a full path indicates a complete rooted path-unit.

3. Pre-processing

The datasets are processed so that only the following features are kept:

1. The OP's title
2. The OP's description
3. The negative comment text
4. The positive comment text

The original paper compares the performances between only the root reply and the full path. Therefore, in replicating the experiment, I also mimic the experiment setting in deriving datasets that only include the root reply or the full path. *(However, I have a question—almost half of my dataset (381) only has root replies and no other replies for a complete rooted path-unit for both positives and negatives. It means that the root reply and full path are the same for these rooted path-units. I'm unsure how the original study did with these rooted path-units, but I included these rooted path-units in the following datasets.)*

In the pre-processing stage, two datasets are created: 1) the first dataset extracts the text with markdown (HTML) and root reply only, and 2) the second dataset extracts the text with markdown (HTML) and full path. The reason behind extracting the HTML instead of the direct text is that the project needs to consider the markdown used in the text.

4. Model

This project used gpt-3.5-turbo-1106 API (due to the low rate limit and high costs for GPT-4). All forms of predictions are obtained purely from prompting the contents in the above datasets to the API.

5. Prompting Approach

The goal of this project is not only to test how LLMs perform in classifying the correct rooted path-unit, but also to understand which features play important roles in such decision processes as the original paper.

The original study used a few sets of features (independently or combined) to fit logistic regressions to predict which full path or root reply wins the OP's delta. These sets include the number of words, BOW, POS, interplay (some features defined in the paper), and style (some features defined in the paper, including four categories).

To mimic the same method through prompting, one straightforward approach would be stating and describing all or one of these sets of features and asking the model to only consider them/it in the decision process, then evaluate the model performances based on what sets of features are described to understand how these features play a role (basically replacing Logistics with LLMs).

Since BOW and POS are mainly word representations, using them in LLMs is unreasonable because LLMs have an inherent knowledge of linguistic features. Therefore, in this replicated project via LLMs, I will only focus on the number of words, interplay, style, and them combined.

However, after an initial experiment with 20 rooted path-units, the performance is stable regardless of what set(s) of features are described. LLM's decision process seems to be unaffected by prompting it only to consider a/all set(s) of features. To provide a better understanding of LLMs' evaluation in the rooted path-unit as well as the features, I used the following approaches:

1. LLMs' Evaluation
    i. Direct evaluation. Directly ask LLM which rooted path-unit changed the OP's opinion.
    ii. Score, then evaluate. Give a score to each rooted path-unit with an explanation, then compare the scores and explanations, and finally make a decision.
2. Features
    i. Do not prompt anything about features; ask what the important features are in the decision process and return the top 5 important features in the text. This can understand how LLMs make decisions for persuasive tasks.

ii. Prompt the features used in the paper and ask independently whether each set of features is being considered in the decision process. This can assess the effectiveness as well as usages of these features. Specifically, the sets are interplay and style. Within style, there are word count, word choices, word emotions, entity-related features, and use of markdown.

In this project, I will first compare which prompting technique would result in the best performance in LLMs, then use that technique to shed insights on the important features in deciding which rooted path-unit changes the OP's opinion from different prompting methods. Please refer the specific prompts in the code (main.py).

Alternate Approach Tried: Comparing to using pure body text (no HTML), the AUC_ROC and accuracy for using HTML body text are both slightly higher (~0.05) for a 50-discussion experiment. Therefore, I omitted the approach to using pure body text (that I thought might be able to improve performance).

Results

1. LLM's Evaluation

Table 1. AUCROC and Accuracy in Evaluating LLMs' Performances

	Direct – Full Path	Direct – Root Reply	Score – Full Path	Score – Root Reply
AUC_ROC	0.566	0.550	0.585	0.567
Accuracy	0.551	0.562	0.584	0.568

Direct indicates the first prompting technique mentioned in the method-prompting approach-LLMs' Evaluation section. Score indicates the second prompting technique. Full path indicates that the dataset 2 described in the method-preprocessing is used, and root reply indicates that the dataset 1 described in the method-preprocessing.

Based on the results above, the "score, then evaluate" prompting method has better AUCROC scores and accuracies for both the full path and the root reply datasets. Within the "score, then evaluate" prompting method, including the full path has better results for both AUCROC scores and accuracies. Therefore, asking LLMs to evaluate the rooted path-units separately with reasonings with all replies is a better approach.

For cost reasons, I will proceed with the "score, then evaluate" prompting method with the full path for the following feature examination.

2. Feature Evaluation

Table 2. AUCROC and Accuracy for Different Feature Prompting

	No Info on Features	Specific Info on Features
AUC_ROC	0.576	0.579
Accuracy	0.574	0.581

No Info on Features indicates the first prompting technique mentioned in the method-prompting approach-Features section. Specific Info on Features is the second prompting technique.

Based on the results above, prompting with specific information of the features used in the original paper seemed to have better overall results. The prompt used for getting the *predictions* are the same for both "No Info on Features" and "Specific Info on Features". The differences might be affected by the additional prompt on the features or model variations.

2.1 "No Info on Features"

Here are the top 20 most frequent important features identified by the LLMs with the "No Info on Features" prompting. In this prompting method, for each discussion, the LLMs give the five most important features in their decision process:
Logical reasoning (117), Personal experience (62), Clarity of argument (32), Empathy (31), Historical context (31), Specific examples (28), Counterarguments (25), Clarity of explanation (15), Different perspective (15), Clarity (15), Relevance to OP's concerns (14), Clarity of expression (14), Relevance (13), Well-structured argument (12), Detailed explanation (11), Emotional appeal (11), Statistical evidence (11), Respectful tone (10), Specific example (10), Insightful perspective (10)

Here is a better illustration through a word could:

Fig 1. Word Could for Feature Frequencies for "No Info" Prompting

Based on the results above, it seems that the "style" (arguments from the challengers) is more important comparing to the "interplay" between the challenger and the OP. "*Relevance to OP's concerns*" only shows up 14 times for 639 discussions.

2.2 "Specific Info on Features"

In “Specific Info on Features” prompting technique, I asked explicitly which sets of features are used. The sets of features include interplay and style. Within style, there are word count, word choices, word emotions, entity-related features, and use of markdown. I will first examine the frequencies of each set comes up in the overall evaluation process, and I will then examine how the AUCROC and accuracies scores differ when style or interplay is considered or not.

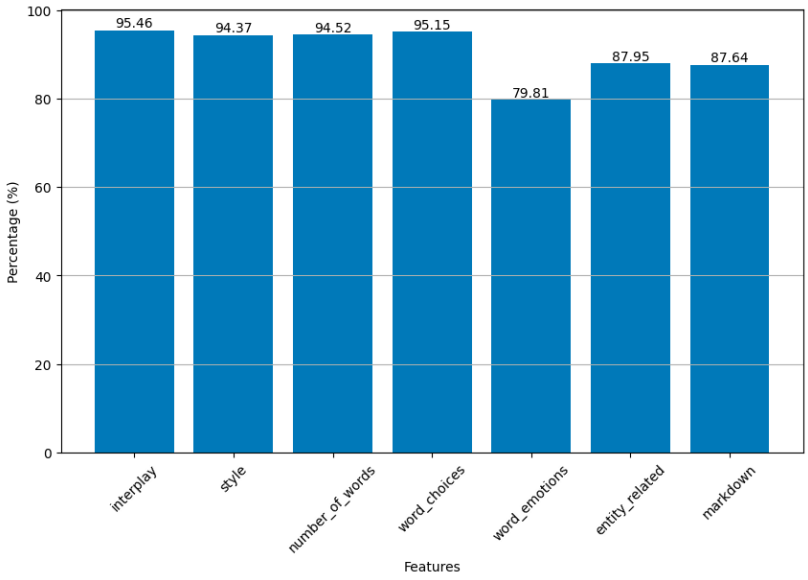

Fig 2. Percentages of Each Feature Being Considered by LLMs

A 95.46 for interplay means that in predicting the results for discussions, interplay is being considered for 95.46% of discussion. It seems that word_emotions (originally word score-based features) is the least considered, followed by entity_related and markdown. Interplay, however, is the most considered.

In the results from the previous “No Info on Features” prompting, interplay does not usually come up in the “five most important features”. However, as it showed up as the most frequent feature here, it might indicate that interplay may only mostly play minor roles.

Table 3. Model performance when “interplay” and “style” is considered or not considered

Feature	AUCROC	Accuracy
Interplay = 0	0.757	0.759
Interplay = 1	0.570	0.572
Style = 0	0.624	0.639
Style = 1	0.576	0.577

When interplay or style is not considered, the performances are better. However, this might be due to the very small sample of interplay = 0 and style = 0.

Yijun Liu

Reflection

Replicating the experiment with LLMs is different from replicating it with other machine learning models because it becomes a prompt-based evaluation and prediction. I tried to replicate the experiment as much as possible, but there must be some differences, particularly in accessing the features. In logistic regressions, the features are straightforward. However, in the prompt, I tried to summarize them with examples for the "Specific Info on Feature" prompting methods. Although I asked the model to label which sets of features were used, it was still hard to assess whether the model was just hallucinating. A future experiment might repeat the experiments and compare the results.

Additionally, I tried another way to understand what features are important directly from the LLMs. It is interesting to see how "interplay" seems to be less relevant in the "No Info on Features" prompting. Additionally, when interplay is not considered for the "Specific Info on Feature," both AUCROC and accuracies go up. Therefore, a future experiment might be prompting the LLMs without the OP's detailed description of the problem. Since the interplay might be less important, omitting the description might provide better results because there will be less information fitted to the LLM, avoiding overfitting.

Another future direction might be refining the sets of features for the second prompting method for features. I mainly adopted these sets of features from the original paper. However, these sets of features might be improved (by reducing or adding features) from additional literature reviews or combining some of them in a different way based on the results from this report. For example, not including interplay.

Compared to the original experiment setting via Logistic Regressions, using GPT 3.5 seems to have worse accuracies and a little bit better AUCROC scores. In the future, it might be worth to try with more advanced models to improve the scores with LLMs. When I was experimenting with my prompts (with 10-15 discussions), it usually classifies around 70% correct with GPT 4 Turbo. However, when I was running on the full dataset, I seemed to be at the limit for GPT 4 Turbo after processing around 600 discussions each day.