Kyvalya Reddy

CSE 428 Final Project Report

Improving Panacea's Patient-Trial Matching Capabilities Using Verification with Other LLMs

Project Topic

Paper Overview

"Panacea: A foundation model for clinical trial search, summarization, design, and recruitment (Lin et al)" is an LLM used for a variety of clinical trial tasks. It was developed by fine-tuning Mistral, a general-knowledge LLM, using the TrialAlign dataset containing trial documents as well as the TrialInstruct dataset containing instructions for specific trial tasks. Panacea enables users to query clinical trials, design a detailed clinical trial protocol, and match patients to clinical trials given their medical history. This improves the efficiency and accessibility of developing and identifying clinical trials in the medical setting. In this project, I focus on improving the accuracy of the patient-trial matching task.

Reproduction of Results

To reproduce the results of the paper, I focused on demonstrating the usage of Panacea in the patient-trial matching task. I loaded the model from Hugging Face, ran it on the first 100 samples in the TREC2021 dataset, and calculated the evaluation metrics of the results. The balanced accuracy was 0.45, which is lower than described in the paper because of the small sample size due to runtime limitations.

Improvement

Goals

My goal was to improve the accuracy of patient-trial matching and reduce LLM hallucinations by verifying results from Panacea with different general-knowledge LLMs that

performed well in patient-trial matching as described in the paper. I planned to evaluate the results of the models and model combinations on the TREC2021 dataset using the evaluation metrics described in the paper.

The input to the problem is patient-trial matching instructions, a description of the clinical trial, and the patient's medical history (all of which are found in the TREC2021 dataset). The output to the problem is whether the patient should be recommended for the trial, where 0 represents "would not refer", 1 represents "would consider referring", and 2 represents "likely to refer".

Methods

I first evaluated four models (Panacea, Mistral, Zephyr, and LLaMA3) on the entire TREC2021 dataset. I used the model outputs of the patient-trial matching task found in the Panacea GitHub rather than running the models on the data, due to runtime limitations. I calculated the evaluation metrics used in the paper, which include balanced accuracy, Cohen's KAPPA, precision, recall, and F1 score.

I then evaluated these models on the Cohort dataset, which was also used in the paper. I used these accuracy results to inform the model combinations described later on in order to avoid data leakage within the TREC2021 dataset.

I formulated four methods of combining the results of the models to improve the accuracy of the prediction. All four methods were evaluated on the TREC2021 dataset and the evaluation metrics were calculated.

The first method is Majority Vote, which involves counting the votes for each output value. If there is a majority (3 out of 4 votes for the same value), return that value. Otherwise,

default to Panacea's output. This method allows us to override Panacea's output if all of the other models agree on a different output, which may reduce Panacea's hallucinations.

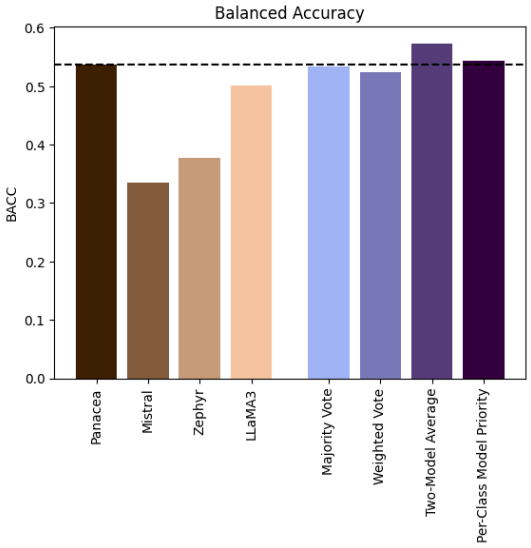
The second method is Weighted Vote, which involves assigning weights to models based on their relative accuracy scores on the Cohort dataset (to avoid data leakage). We calculate the weighted votes for each output value. If there is a majority, we return that value. Otherwise, we default to Panacea's output. This method allows us to override Panacea's output if there is a majority for a different output while ensuring that models with a higher accuracy are prioritized.

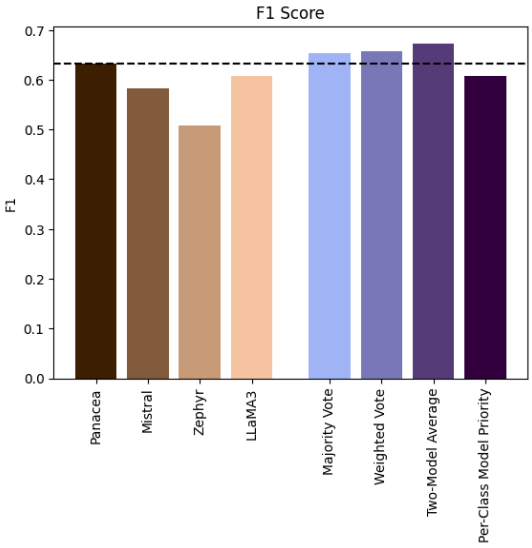
The third method is Two-Model Average, which involves calculating the average of the two models that performed the best on the Cohort dataset (which were Panacea and LLaMA3). We return the floor of this value, rounding down to prioritize safety. This method allows us to rely on only the models that perform well, while still cross-checking Panacea's result with a different LLM.

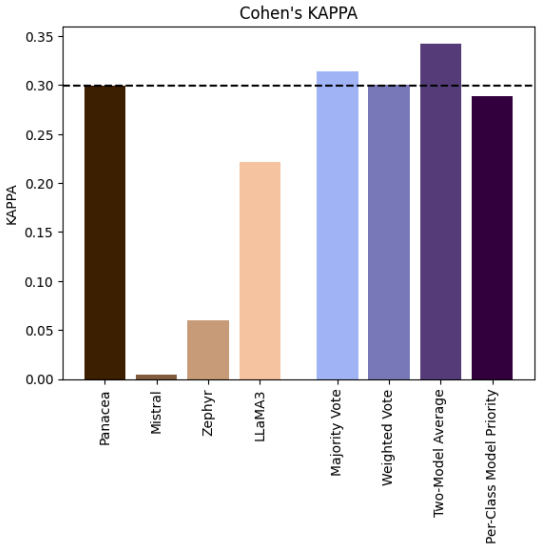
The fourth method is Per-Class Model Priority, which involves prioritizing the results of models that perform better on a certain output value. I first calculated the precision, recall, and F1 score across the four models for each output value to determine if any models perform better on certain outputs. This was done on the Cohort dataset to avoid data leakage. The results from this analysis showed that LLaMA3 had a significantly higher precision and F1 score than Panacea for an output of 2. Thus, if LLaMA predicts 1 or 2 we return LLaMA's output; otherwise we default to Panacea's output. This method allows us to increase the accuracy of particular output values if they prove to be easier for certain models.

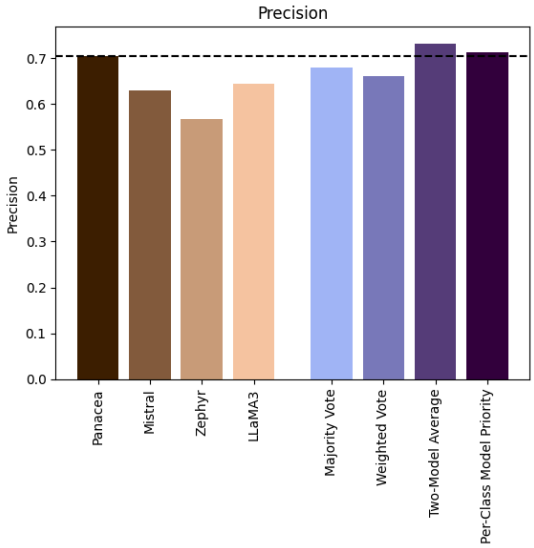
Results

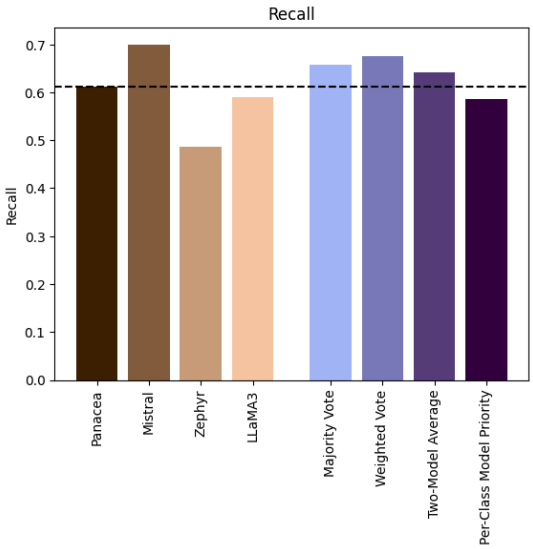
The plots of the evaluation metrics for each model and model combination are shown below.


Balanced Accuracy
BACC
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Panacea
Mistral
Zephyr
LLaMA3
Majority Vote
Weighted Vote
Two-Model Average
Per-Class Model Priority


F1 Score
F1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Panacea
Mistral
Zephyr
LLaMA3
Majority Vote
Weighted Vote
Two-Model Average
Per-Class Model Priority


Cohen's KAPPA
KAPPA
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Panacea
Mistral
Zephyr
LLaMA3
Majority Vote
Weighted Vote
Two-Model Average
Per-Class Model Priority


Precision
Precision
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Panacea
Mistral
Zephyr
LLaMA3
Majority Vote
Weighted Vote
Two-Model Average
Per-Class Model Priority


Recall
Recall
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Panacea
Mistral
Zephyr
LLaMA3
Majority Vote
Weighted Vote
Two-Model Average
Per-Class Model Priority

Two-Model Average was the only model combination that performed better than Panacea across all metrics. Majority Vote and Weighted Vote each performed better than Panacea on 3 out of 5 metrics. Per-Class Model Priority performed better than Panacea on 2 out of 5 metrics but had a slightly higher balanced accuracy than Panacea.

Conclusion

The results show that Two-Model Average performed better than Panacea across all evaluation metrics, which indicates that using a successful general-knowledge LLM such as LLaMA3 to verify the results of Panacea can be helpful in improving performance. The other model combinations improved performance in some of the evaluation metrics, but overall did not appear to be more accurate than Panacea.

Future directions include exploring other ways to combine results from different models, such as fine-tuning the weights in Weighted Vote or assigning different weights to the models based on the predicted output. A more robust method of combining different models could be LLM merging, which involves combining the weights within different models to create one LLM rather than just combining the outputs. Finally, further fine-tuning of Panacea would likely be effective at improving performance.