Oppositional thinking analysis: Conspiracy vs critical narratives

Jere Perisic

[1] University of Southern Maine, Portland ME 04104, USA
[2] jere.perisic@maine.edu

Abstract. Conspiracy theories surrounding public health decisions have arisen in online communication, posing challenges to content moderation and understanding. This paper addresses the distinction between critical and conspiratorial narratives in the context of COVID-19. This study is conducted as part of PAN 2024 and focuses on English and Spanish text corpora. This lab proposes two sub-tasks: binary classifications of critical versus conspiratorial texts, and the identification of elements within oppositional narratives.

Keywords: CLEF · PAN labs · BERT.

1 Introduction

Conspiracy theories have become present in everyday online conversations, particularly in public health discussions, which were emphasized during the COVID-19 pandemic. These narratives are often propagated across various platforms, posing a challenge to moderate and understand the content. Recognizing the distinction is crucial and can bring better content moderation strategies and insight into these conversations.

This paper presents an approach to characterize critical versus conspiratorial texts and the detection of oppositional narrative elements. It utilizes datasets sourced from the Telegram platform and employs SOTA NLP models, including BERT-based classifiers and the Llama-3 model. This paper aims better to understand the mechanisms behind conspiracy theories in online conversations.

2 Related Works

Subtask 1 of this lab has already been part of this research. A couple of papers have researched it. Subtask 2 is newly introduced as part of Oppositional thinking analysis.

In earlier work, there has been the creation of a ConspiDetector, which utilizes CNNs to detect conspiracies[4]. Peskine et al. utilized TFIDF and transformer-based models[6]. Also, work has been done utilizing GPT-3 and prompting[7]. And almost every paper has done some work with CT-BERT, a BERT-based model pre-trained on COVID-19 related Tweets.

3 Models & Research Question

In this section, we will discuss the models used for PAN 2024.

3.1 covid-twitter-bert-v2

CT-BERT is a transformer-based model pre-trained on a corpus of COVID-19-related Twitter messages. It's a domain-specific model, and it shows 10-30% improvement compared to its base model, BERT-LARGE. It is best used on COVID-19 content.

3.2 twitter-xlm-roberta-base-sentiment

This model is designed to handle the unique characteristics of Twitter content. The base model used is XLM-RoBERTa, a RoBERTa-based model trained on multilingual data. It is also based on transformer architecture and pretrained on a large corpus of Twitter data.

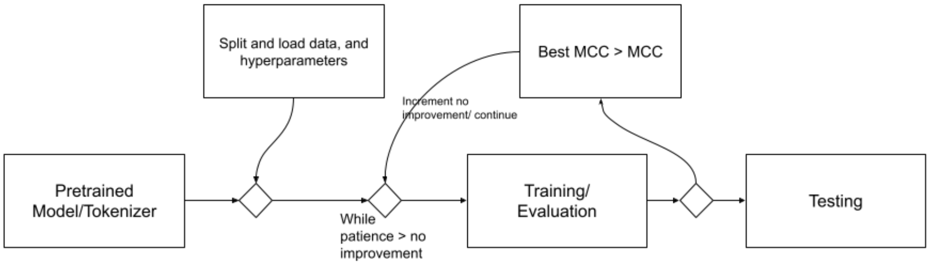
3.3 Meta-Llama-3-8B-Instruct

This Llama model has 8 billion parameters, and it was released on April 18^{th}, 2024. It showed SOTA performance across a wide range of NLP tasks. Llama-3 has increased its vocabulary to 128,256 tokens and has a context length of 8,000 tokens. The data used to train this model had a diverse set of languages.

3.4 Implementation

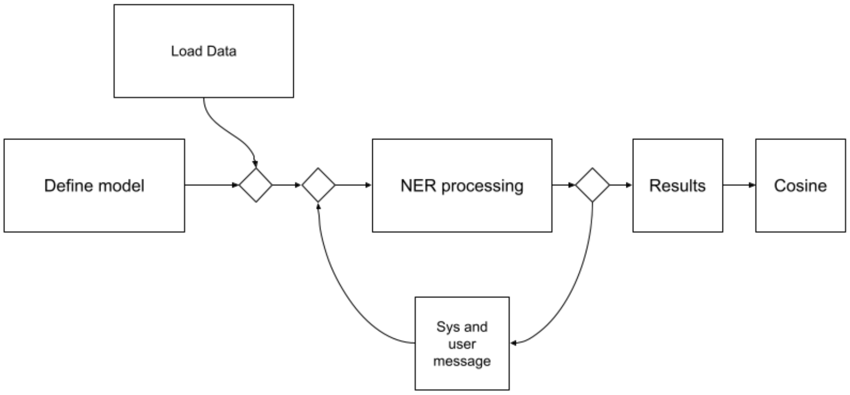
covid-twitter-bert-v2 was implemented for binary classification using PyTorch and the Hugging Face Transformer library.

1. This model was trained using cuda-enabled NVIDIA GeForce RTX 3060 Ti
2. It loads a pre-trained CT-BERT model and tokenizer from Hugging Face.
3. Data is split 80/10/10
4. Training loop
   - In each epoch, sets the model to training mode
   - Processes each training example computes loss and does backpropagation
   - At the end of each epoch, it evaluates the model and saves the best one based on MCC measurement.
   - If the current validation is lower than the previous, the model doesn't get updated, and no improvement counter increments
   - If current validation is higher, the new model is saved, and no counter is incremented
5. The testing loads the best model and calculates MCC and confusion matrix based on testing data.
6. **twitter-xlm-roberta-base-sentiment** has the same architecture


Split and load data, and hyperparameters
Best MCC > MCC
Increment no improvement/ continue
Pretrained Model/Tokenizer
While patience > no improvement
Training/ Evaluation
Testing

Llama-3 performs Named Entity Recognition (NER) using a pre-trained language model.

1. Define model and load the data from JSON file
2. NER processing
   - Iterate through JSON data
   - Extract ID and text from each item
   - System and user messages, forming prompt for a pipeline
   - Generate NER output
   - Parse generated text to extract the output
   - write extracted elements to output JSON
3. Cosines and evaluation of Llama is done separately.



3.5 Data

PAN lab provided us with data for this task. It has two JSON files with telegram posts: one dataset in Spanish and one in English.

Data Example

```
{
    "id": "11414",
    "text": "\" COVID : ALLEGED DOCUMENT PREDICTING
    THE FRENCH STRATEGY IN 2021
    Picture translated to English . ",
    "category": "CRITICAL",
    "annotations": [
      {
        "span_text": "People",
        "category": "VICTIM",
        "annotator": "gold_label",
        "start_char": 943,
        "end_char": 949,
        "start_spacy_token": 177,
        "end_spacy_token": 178
      }
    ],
```

3.6 Evaluation

We used two evaluation metrics for this project: Matthews Correlation Coefficient and F1 score.

Matthews Correlation Coefficient is a statistical method for measuring the quality of binary classifications in machine learning. It is a single-value metric that utilizes a confusion matrix. The confusion matrix consists of four variables, true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP) \cdot (FN + TN) \cdot (TP + FN) \cdot (FP + TN)}}$$

F1 score is a metric used in machine learning to measure models' reliability by calculating the harmonic mean of precision and recall.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

McNemar's test is a statistical test commonly used on confusion matrices. It is used to determine if there is a significant proportion difference between two related groups.

$$x^2 = \frac{(b - c)^2}{b + c}$$

3.7 Research Questions

Hypothesis

- H_0 : There is a significant difference in the performance of the fine-tuned model compared to the baseline model.
- H_1 : There is no significant difference in the performance of the fine-tuned model compared to the baseline model.

Questions

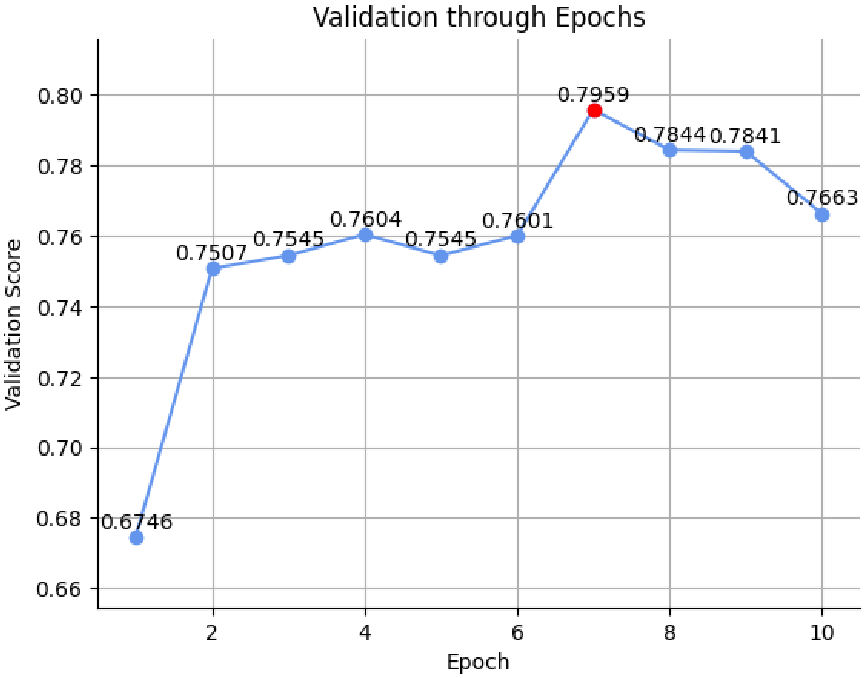
1. Is there a significant difference in the performance of fine-tuned compared to baseline?
2. If a significant difference is detected, which model performs better, and in what aspects?

4 Result

In this section, we will discuss the results for Subtask 1 and Subtask 2.

4.1 covid-twitter-bert-v2

To better the baseline scores, I have decided to fine-tune CT-BERT and introduce early stopping. The patience level has been set to 3 and the best model is saved. For the testing of this model, the best model from early stopping is loaded.



From this plot, we can see that the model went through ten epochs, and the seventh model had the best validation. We saved the model from the seventh epoch and used it for the testing. Results in table 1.

Model	MCC
Baseline	0.0226
Fine-Tuned	0.7554

Table 1. Comparison of baseline and fine-tuned model

We can see that the fine-tuned model drastically outperforms the baseline model.

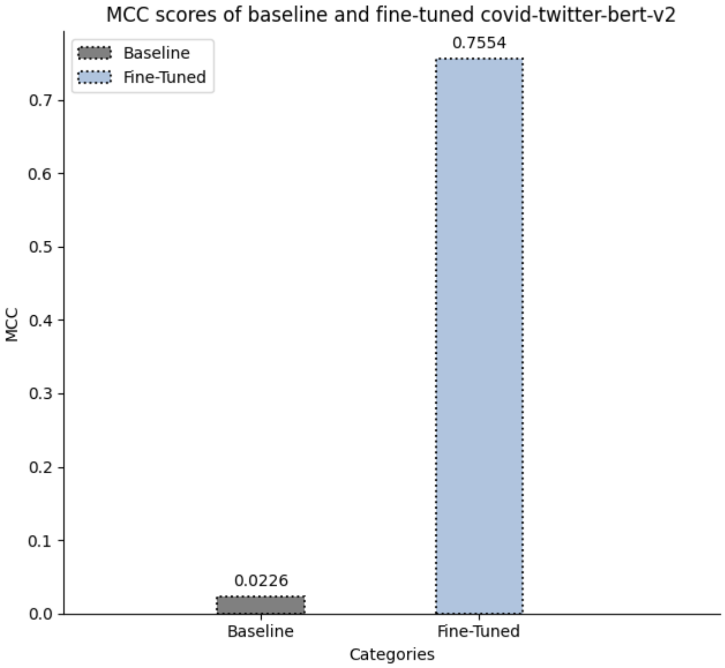


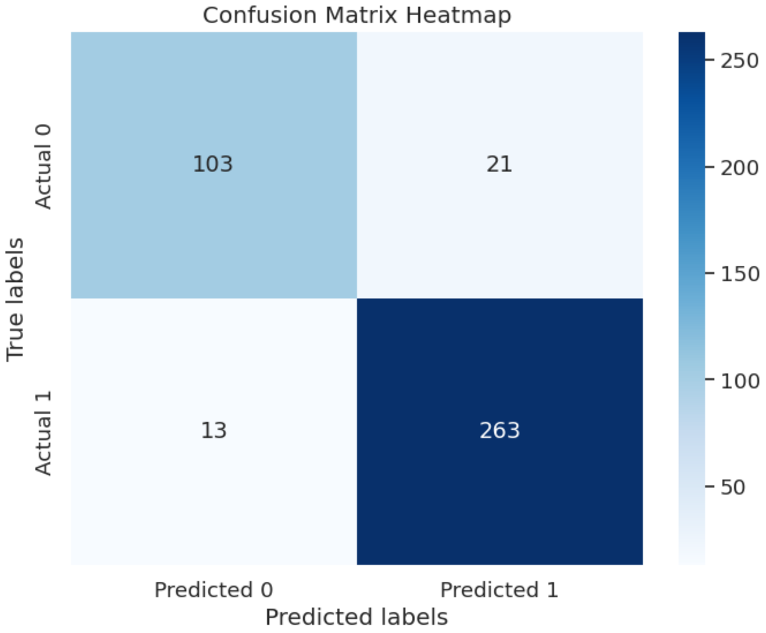
Fig. 1. COVID-Twitter-Bert comparison

Example of success The model predicted conspiracy correctly.

```
THIS IS MASSIVE Australian Senator Malcolm Roberts exposes nanotech
found in the Covid  vaccines  and says they are genocide
He is the first politician to expose this
gold_label: CONSPIRACY,
prediction: CONSPIRACY
```

Example of failure Model mistakes conspiracy for critical thinking.

```
Joe Biden  who told Americans last year You re not going to get covid
if you have these vaccinations has just tested positive for COVID after having
had four vaccinations
gold_label: CONSPIRACY,
prediction: CRITICAL
```



From this confusion matrix, we can see that the model performs overall great when it comes to true positives and true negatives. Only ≈ 8.5% of the data is labeled as false negative or false positive. However, in this case, it is better to have false negatives than false positives.

In the case of covid-twitter-bert-v2, *null hypothesis* is satisfied, and based on the significant testing, the fine-tuned model performs significantly better than the baseline model.

4.2 twitter-xlm-roberta-base-sentiment

This model has a task similar to CT-BERT; the only difference is that it works with the Spanish corpus and is not specifically fine-tuned on COVID-19-related Twitter corpus but a diverse one. In this model, we have also implemented early stopings.

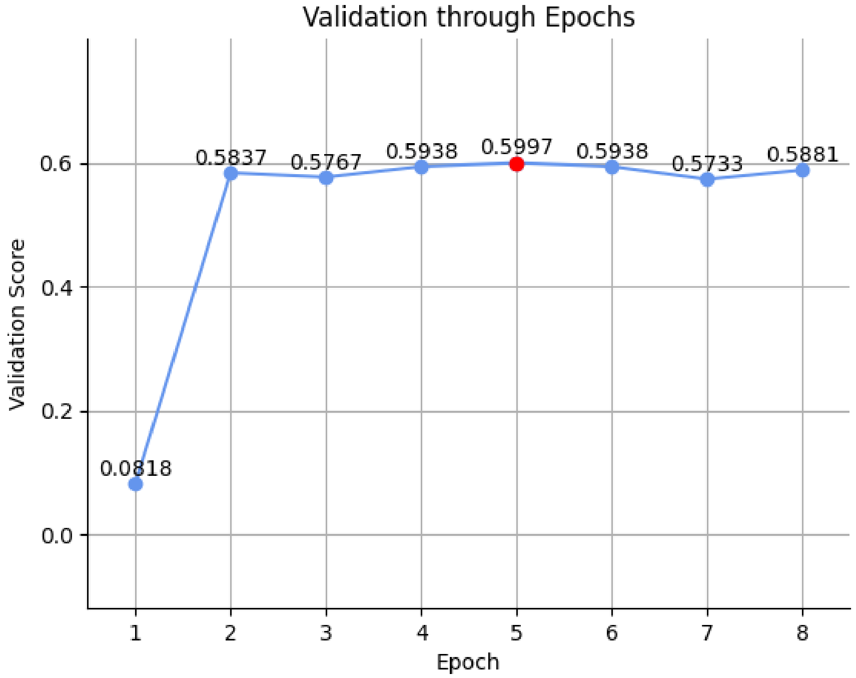
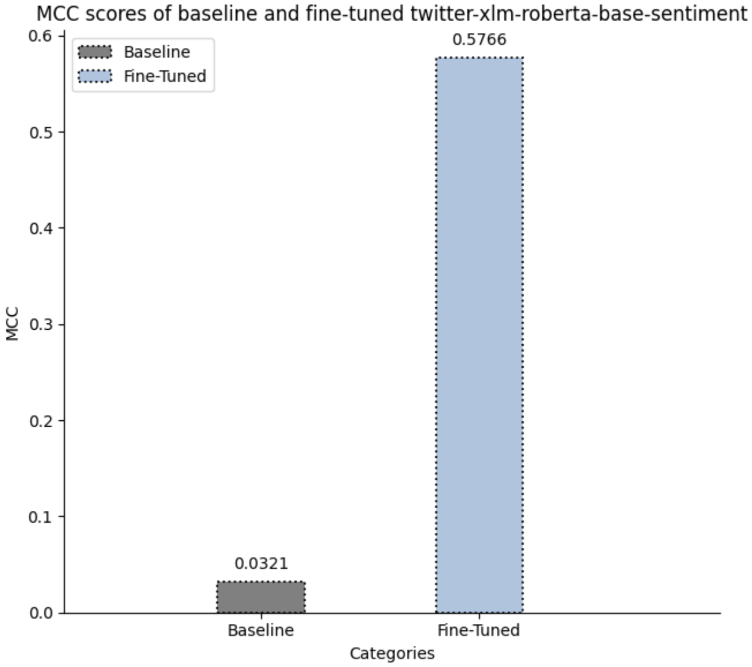


Fig. 2. Scatter plot for the twitter-xlm-roberta-base-sentiment model

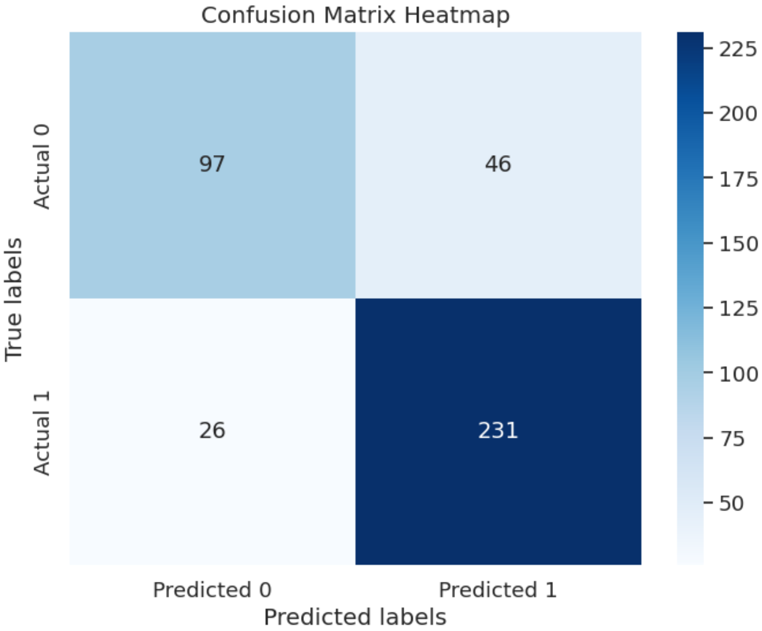
In the figure 2, we can see that the first validation was slightly better than a baseline model. This only happened in this run; after the first run, the data increased to epoch five, where it reached the best result. That model is saved an used for the testing. I chose this plot because it emphasizes the importance of epochs and re-running your model. The table 2 shows the difference between the fine-tuned model and the baseline.

Model	MCC
Baseline	0.0321
Fine-Tuned	0.5766

Table 2. Comparison of baseline and fine-tuned model



This fine-tuned model performs well with Twitter data in Spanish because it is a cross-lingual model trained on the diverse Twitter corpus.



From the confusion matrix, we can see that the model performs well when classifying true positives and true negatives, but there is a noticeable decrease in the performance of mislabeling; in this case, ≈ 18% of the data is mislabeled.

Based on significant testing, the *null hypothesis* is satisfied, and this model is significantly better than the baseline model, but it's not significantly better than CT-BERT.

4.3 Llama-3

In the case of Llama-3, I have experimented with a few prompts, but the best results came from the following prompt.

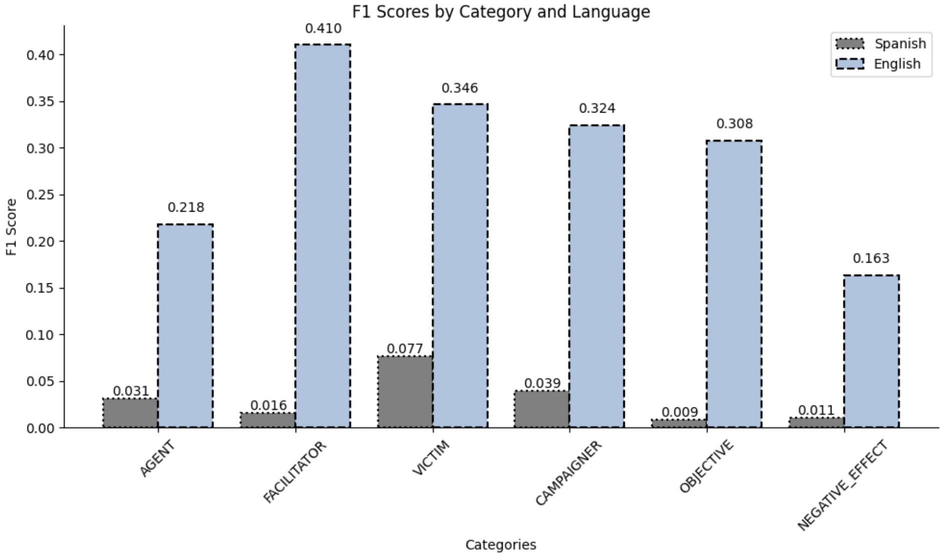
```
Task: Extract specific elements-AGENT, FACILITATOR, VICTIM, CAMPAIGNER,
OBJECTIVE, and NEGATIVE_EFFECT-from a given text, omitting any elements
that are not explicitly mentioned. Instructions: Provide the identified
elements directly from the input text without alterations. Only include
elements that are mentioned explicitly.
```

The prompt was also translated into Spanish. The results generated by this prompt have been parsed and saved into JSON. The problems I have encountered with Llama are that sometimes the model wouldn't output data, and the evaluation metric doesn't provide how good the predictions are. This section requires more work.

The data collected into JSON is first compared with the given dataset, and cosine similarity between the two is created; if the cosine similarity is above a certain threshold, then the given output is considered as predicted correctly. A separate evaluation file calculates precision and recall and, based on that, calculates the F1 score.
The following scores are recorded for Llama-3 and the prompt mentioned above.

Corpus	AGENT	FACILITATOR	VICTIM	CAMPAIGNER	OBJECTIVE	NEGATIVE_EFFECT
English	0.2177	0.4101	0.3462	0.3237	0.3076	0.1633
Spanish	0.0314	0.0161	0.0771	0.0393	0.0089	0.0112

Table 3. F1 scores for each Named Entity



We can see that the English corpus performs worse than guessing randomly, while the Spanish corpus performs almost as never guessing correctly. Problems with the evaluation method include Llama-3 straying away from the prompt or producing different outputs. More time should be spent adjusting the prompt and analyzing the output data. Sometimes, the meaning of predicted output is similar to the gold_label, but cosine similarity is not above the set threshold.

Examples of failed predictions

```
{
    "attribute": "NEGATIVE_EFFECT",
    "dataset_value": "I \u2019m deeply concerned that the push
    to vaccinate these children is nothing more than a dystopian
    experiment with unknown consequences",
    "prediction_value": "Unknown consequences",
    "cosine_similarity": 0.0
 }
```

We can see that the prediction of NEGATIVE_EFFECT is correct, but it didn't include everything and thus is marked as incorrect with a value of 0.0.

```
{
    "attribute": "OBJECTIVE",
   "dataset_value": "None mentioned",
   "prediction_value": "to analyze blood samples under an optical microscope",
    "cosine_similarity": 0.0
},
```

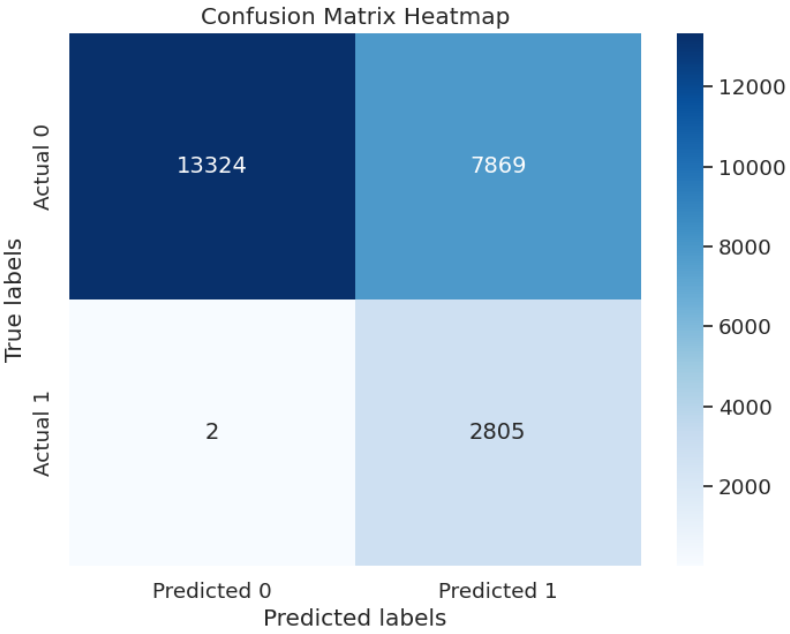
In this dataset, there is no OBJECTIVE, but the Llama-3 predicted an OBJECTIVE.

Example of successful predictions

```
    {
        "attribute": "VICTIM",
        "dataset_value": "someone who died suddenly",
        "prediction_value": "someone who died suddenly",
        "cosine_similarity": 1.0
    },
    {
        "attribute": "CAMPAIGNER",
        "dataset_value": "None mentioned",
        "prediction_value": "none mentioned",
         "cosine_similarity": 1.0
    },
```

In both cases, Llama-3 predicted correctly. In the first case, it predicted correct VICTIM, and in the second case, it correctly predicted CAMPAIGNER by not predicting.

The confusion matrix of the Llama-3 model is strange, and there is a high chance of a coding issue; I have made a code to generate both F1 scores based on cosine similarity and confusion matrix.



In this confusion matrix, we can see that there is a high number of true positives and true negatives. While there is a small and almost nonexistent number of false negatives, there is a $\approx 32.8\%$ of false positives. The issue of why that happens can be from parsing the data to evaluating the files. Also, it can be that the model is generating false positives.

Although my baseline model for this was faulty and nonperformance, I still did significant testing, and it shows that both Llama-3-8b-Instruct that uses Spanish and English data are significantly better.

5 Conclusion

A fine-tuned model based on COVID-twitter-bert-v2 performs best since the model before it is domain-specific and pre-trained on a large corpus of COVID-19-related data. I expected a better result from twitter-xlm-roberta-base-sentiment since it's a model trained on Twitter data, but I think the dataset provided needs to be reevaluated. My biggest concern is the data provided and its validity; I wish I had a chance to examine it properly and then test my models on that data. I am satisfied with the results provided by Llama-3; I think the actual scores are higher, but my knowledge limited me to examining Llama-3 predictions correctly.

I think the next steps are to check the validity of the data, remove all the languages that do not belong to that dataset, and re-run the models. Also, predictions from Llama-3 need to be examined in depth, and proper metrics must be used to evaluate the outputs. I have used the F1 score since it is the official metric used by the PAN lab.

References

1. Barbieri, F., Anke, L. E., and Camacho-Collados, J. Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond. *arXiv preprint arXiv:2104.12250* (2021).
2. Chicco, D., and Jurman, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics 21* (2020), 1–13.
3. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
4. Giachanou, A., Ghanem, B., and Rosso, P. Detection of conspiracy propagators using psycho-linguistic characteristics. *Journal of Information Science 49*, 1 (2023), 3–17.
5. Müller, M., Salathé, M., and Kummervold, P. E. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503* (2020).
6. Peskine, Y., Alfarano, G., Harrando, I., Papotti, P., and Troncy, R. Detecting covid-19-related conspiracy theories in tweets. In *MediaEval* (2021).

7. Peskine, Y., Korenčić, D., Grubisic, I., Papotti, P., Troncy, R., and Rosso, P. Definitions matter: Guiding gpt for multi-label classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023* (2023), pp. 4054–4063.