NLPDame at ClimateActivism 2024: Mistral Sequence Classification with PEFT for Hate Speech, Targets and Stance Event Detection

Christina Christodoulou
Institute of Informatics & Telecommunications,
National Centre for Scientific Research, "Demokritos"
Athens, Greece
ch.christodoulou@iit.demokritos.gr

Abstract

The paper presents the approach developed for the *Climate Activism Stance and Hate Event Detection* Shared Task at CASE 2024, comprising three sub-tasks. The Shared Task aimed to create a system capable of detecting hate speech, identifying the targets of hate speech, and determining the stance regarding climate change activism events in English tweets. The approach involved data cleaning and pre-processing, addressing data imbalance, and fine-tuning the *mistralai/Mistral-7B-v0.1* LLM for sequence classification using PEFT (Parameter-Efficient Fine-Tuning). The LLM was fine-tuned using two PEFT methods, namely LoRA and prompt tuning, for each sub-task, resulting in the development of six Mistral-7B fine-tuned models in total. Although both methods surpassed the baseline model scores of the task organizers, the prompt tuning method yielded the highest results. Specifically, the prompt tuning method achieved a Macro-F1 score of 0.8649, 0.6106 and 0.6930 in the test data of sub-tasks A, B and C, respectively.

1 Introduction

Climate change is an ever-growing concern that has garnered significant attention worldwide. As the severity of its impacts becomes increasingly undeniable, it has also become an issue that has sparked diverse reactions and discussions on social media platforms. Within these discussions, the prevalence of hate speech, the identification of its targets, and the detection of various stances towards climate change and activist movements have become vital areas of interest. Understanding the dynamics of hate speech, targets of hate speech, and different stances within climate change discourse is crucial for fostering informed discussions, addressing concerns, and promoting positive change. Hate speech, defined as harmful or offensive language directed towards individuals or groups, has the potential to exacerbate division, hinder productive conversations, and impede constructive collaboration. Identifying hate speech in climate change discourse provides a deeper understanding of the negative impact it can have on the overall conversation. Additionally, recognizing the targets of hate speech helps shed light on the specific groups or entities facing hostility, enabling targeted interventions and support. Examining the different stances towards climate change and activist movements also unveils the diversity of perspectives within these discussions. Stance detection allows for the identification of supporters, skeptics, and deniers, providing a nuanced understanding of the range of viewpoints on this pressing issue. By capturing shifts in opinions, trends can be identified, informing future discussions and policy-making.

Natural Language Processing (NLP) models have proven to be valuable assets in detecting hate speech, determining its targets, and classifying stances within various domains. However, when it comes to climate change discourse, there is a need for well-annotated datasets that specifically address the unique challenges present in this field. The scarcity of such datasets poses a significant obstacle to harnessing NLP models effectively. To address this gap, Thapa et al. (2024) created the *Climate Activism Stance and Hate Event Detection* Shared Task at CASE 2024 which challenged participants to develop binary and multi-class text classification systems that are able to detect hate speech, targets of hate speech as well as stance detection concerning climate change, events and movements. The Shared Task leveraged several aspects of the annotated English Twitter dataset regarding climate discourse made by Shiwakoti et al. (2024). This paper presents the system developed for this Task, with the code available on the provided GitHub link.[1]

[1] https://github.com/christinacdl/Climate_Activism_Stance_and_Hate_Event_Detection_CASE_

Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024), pages 96–104

The structure of this paper is as follows: Firstly, Section 2 presents a discussion of the previous related work followed by the presentation of the task and data analysis in Section 3, and an overview of the developed methodology in Section 4. Section 5 presents the results and error analysis. Finally, the paper concludes with Section 6, which discusses future work, as well as the limitations during participating in the Task.

2 Related Work

As social media usage continues to grow and user-generated content becomes more prevalent, numerous studies have focused on identifying and categorizing insulting messages that target individuals or groups across different platforms. To accomplish this, researchers have utilized NLP in conjunction with machine learning. While initial studies focused solely on the English language, the need to address this issue in a multi-lingual context has emerged in recent years. Many studies and shared tasks have been conducted, utilizing various terms such as abuse, aggression, cyberbullying, hate speech, and toxic or offensive language to classify these messages. SemEval's 6^{th} shared task, OffensEval: *Identifying and Categorizing Offensive Language in Social Media*, introduced the detection of offensive language on social media. The task consisted of three sub-tasks that aimed to implement binary or multi-class text classification. Sub-task A sought to differentiate between offensive and non-offensive English tweets, while sub-task B aimed to identify the type of offensive tweets and whether they were targeted or not. Sub-task C aimed to identify the target of the offensive posts. Participants were provided with a dataset containing 13,240 English tweets and a test set of 860 tweets, called the *Offensive Language Identification Dataset* (OLID), which were annotated according to the three sub-tasks (Zampieri et al., 2019). This task was extended the following year as the 12^{th} task of SemEval 2020 named as *Multilingual Offensive Language Identification in Social Media* to encourage offensive language detection in other languages, such as Arabic, Danish, Greek and Turkish, based on the sub-tasks of the previous SemEval (Zampieri et al., 2020). Moreover, SemEval's 5^{th} task in 2019 addressed the issue of hate speech directed towards immigrants and women on Twitter, in both English and Spanish. The two sub-tasks required binary classification - indicating whether a post was hateful or not - and determining whether the target was a generic group or an individual (Basile et al., 2019). In addition, Gautam et al. (2019) analyzed 9,973 tweets related to the *MeToo* movement. They identified five dimensions: stance, relevance, hate speech, dialogue acts, and sarcasm. This analysis provided valuable insights into how people use language to discuss sensitive social issues like *MeToo* on social media platforms. Nevertheless, Parihar et al. (2021) released a paper that discussed the challenges in hate speech detection, including the subjective nature of annotations and the lack of language models for regional languages. Despite the great endeavour in mitigating hate speech and dealing with various social issues, there remains a significant gap in the study of climate change discourse, particularly in the analysis of climate discourse on social media platforms from multiple perspectives. In their efforts to advance this field, Webersinke et al. (2021) introduced *ClimateBERT*, a domain-specific LM that was trained on a staggering 2,046,523 climate-related paragraphs. Additionally, Stammbach et al. (2023) curated a dataset of 3,000 binary datasets focused on environmental claims, often made by businesses in the finance sector. As per their experiments, transformer models have outperformed non-neural models.

3 Task & Dataset

3.1 Task

The identification of hate speech and stance detection are critical components in recognizing events that occur during climate change activism. In order to detect hate speech, it is essential to identify the occurrence of hate speech as the event, the entity as the target of the hate speech, and the relationship between the two. The identification of targets is a crucial task in hate speech event detection. Furthermore, stance event detection is a vital part of comprehending whether activist movements and protests related to climate change are being supported or opposed. The Shared Task at CASE 2024 aimed to address these issues and was divided into three sub-tasks: detection of hate speech (sub-task A), targets of hate speech (sub-task B), and stance (sub-task C). More particularly, sub-task A, Hate Speech Detection, involved identifying whether a given text contains hate speech or not. The text dataset for this sub-task consisted of binary annota-

2024.git

tions for the prevalence of hate speech. Sub-task B, Targets of Hate Speech Detection, involved identifying the targets of hate speech in a given hateful text. The text was annotated for *individual*, *organization*, and *community* targets. Finally, sub-task C, Stance Detection, involved identifying the stance in a given text. The text was annotated for three different stances: *support*, *oppose*, and *neutral*. Hence, sub-task A required binary text classification, while sub-task B and C required multi-class text classification (Thapa et al., 2024).

3.2 Dataset

The provided dataset was created by Shiwakoti et al. (2024) who collated 15,309 English tweets related to climate change, events, and activist movements posted during the year 2022 using the Twitter API. They employed relevant hashtags, including #climatecrisis, #climatechange, #ClimateEmergency, #ClimateTalk, #globalwarming, #fridaysforfuture, #actonclimate, #climatestrike, #extinctionrebellion, #ClimateAlliance, #climatejustice, and #climateaction to retrieve the tweets. The tweets were then annotated for various aspects, such as relevance, stance, humor, hate speech as well as direction and targets of hate speech.

The training data for sub-tasks A and C consisted of 7,284 tweets. In comparison, the validation data included 1,561 tweets. The test data comprised 1,562 tweets. For sub-task B, the training data amounted to 699 tweets, while the validation and test data had 150 tweets each. While cleaning the data, it was discovered that all data sets contained duplicate tweets. The training data had 365 duplicate tweets, while the validation and test data had 33 and 47 duplicate tweets, respectively, for sub-tasks A and C. For sub-task B, the training data had 237 duplicate tweets, while the validation and test data had 18 and 31 duplicate tweets, respectively. To ensure data uniformity, only the first occurrence of each tweet was retained in the training and validation datasets. However, no duplicates were removed from the test data to ensure the final evaluation of the system was not affected. The training data was used only for training, no data splitting was applied for evaluation. The class distribution of the three training sets before and after data cleaning as well as the categorical labels, along with their respective numerical labels provided by the organizers, are presented in Table 1. From the training data, it became evident that several classes, namely *HATE*, *COMMUNITY*, and *OPPOSE* in sub-task A, B and C, respectively, were under-represented and formed the minority of the classes. For this reason, different weights were assigned to the loss function for each class providing higher weight to the minority classes and lower weight to the majority classes. Although the labels of all the validation and test sets were provided after the end of the evaluation and testing phases, it became evident that their class distribution was consistent with the class distribution of the training set.

Class Label	Before Data Cleaning	After Data Cleaning
Sub-task A		
NON-HATE (0)	6,385	6,262
HATE (1)	899	657
Sub-task B		
INDIVIDUAL (1)	563	326
ORGANIZATION (2)	105	105
COMMUNITY (3)	31	31
Sub-task C		
SUPPORT (1)	4,328	4,246
OPPOSE (2)	700	458
NEUTRAL (3)	2,256	2,215

Table 1: Categorical & Numerical Labels with Class Distribution in Training Sets.

4 Methodology

4.1 Mistral LLM & PEFT Methods

Mistral is a 7-billion-parameter language model that has been designed to deliver high performance and efficiency in text generation (Jiang et al., 2023). It utilizes grouped-query attention (GQA) to ensure faster inference and sliding window attention (SWA) to handle long sequences effectively. The model has been evaluated and outperforms the Llama 2 13B model across all benchmarks. It also outperforms the Llama 1 34B model in reasoning, mathematics, and code generation. The model's architecture is based on a transformer with specific parameters such as a window size of 4096 and a context length of 81,922. It is available on Hugging Face under the name *mistralai/Mistral-7B-v0.1* for easy deployment and fine-tuning across various tasks.[2] There are also

[2] https://huggingface.co/mistralai/Mistral-7B-v0.1

two instruct versions of Mistral (*mistralai/Mistral-7B-Instruct-v0.1* and *mistralai/Mistral-7B-Instruct-v0.2*) which were fine-tuned using a variety of publicly available conversation datasets. To leverage for fine-tuning, they require surrounding the prompt with the *[INST]* and *[/INST]* tokens. After careful consideration, it was decided that the Mistral base model architecture would be the sole focus of the presented approach, even though there was the possibility of using more LLMs for experimentation and comparison. The decision was based on the understanding that the Mistral base model offered a solid foundation for evaluating text generation performance, and it would be interesting to assess its text classification performance as well. Additionally, assessing multiple models could detract from the accuracy and clarity of the results. Therefore, it was determined that a focused approach would be more effective in achieving the research objectives.

The PEFT library, which is integrated with Hugging Face's Transformers, includes methods that are designed for the efficient adaptation of large pre-trained models to various downstream applications. These methods enable fine-tuning a small subset of additional model parameters, which helps in reducing the computational and storage demands.[3]

LoRA (Low-Rank Adaptation) is one of the PEFT methods which adapts LLMs to specific tasks while reducing the number of trainable parameters (Hu et al., 2021). This method freezes pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture. This significantly reduces the number of parameters that need to be trained, making it more efficient in terms of memory and storage usage. With LoRA, LLMs allow efficient task switching while reducing hardware requirements for training. Moreover, LoRA introduces no additional inference latency compared to fully fine-tuned models. Empirical investigations have shown that LoRA performs on par or better than fine-tuning on various models like RoBERTa and DeBERTa, suggesting that it amplifies important features for specific downstream tasks that were learned but not emphasized during general pre-training. To fine-tune a model using LoRa, the task type, the dimension of the low-rank matrices (LoRA r), the scaling factor for the weight matrices (LoRA alpha), and the dropout probability of the LoRA layers (LoRA dropout) as well as the LoRA bias to train all bias parameters needed to be defined. For the present approach, the selected task type was *SEQ_CLS* and the default LoRA dropout was used. The same number was set for r and alpha as a starting point as was suggested because it is very easy to reduce the impact of LORA data after the training, in case it appears to be too dominant and overtakes the entire model.[4]

Prompt tuning is a technique used to adapt large pre-trained language models for specific downstream tasks by learning *soft prompts* that are added to the input text (Lester et al., 2021). These soft prompts are learned by backpropagation and can incorporate signals from labelled examples. This is different from the discrete text prompts used by models. The main advantage of prompt tuning is that it allows for the reuse of a single frozen model across multiple tasks, which is more efficient in terms of storage and computational resources compared to traditional model tuning where all model parameters are adjusted. The effectiveness of prompt tuning is demonstrated by its ability to outperform few-shot learning approaches like GPT-3's prompt design and to match the strong performance of model tuning as the size of the language model increases. It also shows improved robustness to domain shifts, suggesting that it can help avoid overfitting to specific domains. After creating multiple prompts, Table 2 displays the final versions of the prompts that were created using this method for each sub-task. During experimentation, it was revealed that Mistral performs better when the *[INST]* and *[/INST]* tokens are added at the beginning and end of the prompt. Thus, it appears that the Mistral base model closely resembles its instruction models during prompt construction.

4.2 Environment Setup

The presented methodology was implemented in three separate Python files, one dedicated to each sub-task. The experiments were mainly conducted using the *Transformers*, *PEFT* and *Hugging Face* libraries and 1 NVIDIA RTX, 24210.125MB. The model was loaded in 4-bit Quantization using the *BitsAndBytesConfig* library which is integrated with Hugging Face. Quantization was used to reduce memory usage and speed up model execution while maintaining accuracy.

[3] https://huggingface.co/docs/peft/index

[4] https://medium.com/@fartypantsham/what-rank-r-and-alpha-to-use-in-lora-in-llm-1b4f025fd133

Text Prompt
Sub-task A
[INST]Your task is to classify if the text contains hate speech or not, and return the answer as the corresponding label '0' or '1'[/INST]
Sub-task B
[INST]Your task is to classify the target of hate speech as individual, organization or community, and return the answer as the corresponding label '0' or '1' or '2'[/INST]
Sub-task C
[INST]Your task is to classify the stance of hate speech as support, oppose or neutral, and return the answer as the corresponding label '0' or '1' or '2'[/INST]

Table 2: Text Prompts created for prompt tuning with Mistral-7B in each sub-task.

4.3 Pre-processing & Hyperparameters

Several pre-processing steps were applied to the tweets of all training, validation, and test sets using a function that included regular expressions and other functions. Firstly, all emojis were converted to their textual representations (Taehoon et al., 2022).[5] The *&* and *&* were replaced with *and*. The ASCII encoding apostrophe was replaced with the UTF-8 encoding apostrophe. Consecutive non-ASCII characters were replaced with whitespace, and all extra whitespace was removed. Then, the python *wordsegment*[6] library as well as the *Ekphrasis* library were leveraged for hashtag segmentation (Baziotis et al., 2017).[7] The *Ekphrasis* library was also employed for normalizing the usernames, links and emails by converting them into the special tokens <*user*>, <*url*> and <*email*>, respectively. They were selected to be anonymized for data privacy. They were not removed completely, instead, they were replaced by the aforementioned special tokens to avoid any loss of context. Removing the usernames, especially in sub-task B whose aim is to detect the hate speech target, would result in great loss of performance. Finally, the case and punctuation were maintained as they contribute to the context of the text.

Following the pre-processing steps, the training, validation and test data were converted from dataframes into JSON datasets. The datasets were passed to the LLM's tokenizer, which tokenized and returned the tweets into input IDs and attention masks. The train, validation and test datasets were concatenated for each sub-task to get the overall maximum sequence length of the input IDs, which was employed in each sub-task and is shown in Table 7 of Appendix A along with all hyperparameters. Identical hyperparameters were employed for both LoRA and Prompt Tuning models in each sub-task to ensure consistency and easy model comparison. Only one specific random seed (42) was selected during fine-tuning across all experiments of sub-tasks to ensure reproducibility.

To address the data imbalance, the weight of each class was calculated based on the ratio of the total number of training samples to the number of training samples in that class. These weights were then passed into the *CrossEntropy Loss* function. This approach ensured that classes with fewer samples had a higher weight, whereas classes with more samples, which were over-represented in the dataset, had a lower weight during fine-tuning. At this point, it is important to note that the labels in sub-tasks B and C were converted from 1,2,3 to the corresponding integers 0,1,2 for fine-tuning the LLM. The correct labels were assigned during the creation of the submission files. In Table 6 of Appendix A, the calculated weights for each class in each sub-task are presented.

The system's efficiency and final ranking were primarily evaluated based on the Macro-F1 score of the test set predictions. Thapa et al. (2024), the task organizers, had released their fine-tuned models as baselines along with their Macro-F1 and accuracy scores for each task, which were employed for comparison with the approach presented in this paper in Table 4. Finally, the Macro-F1 score for each class and Confusion Matrices were calculated for error analysis.

5 Results & Discussion

Table 3 shows that Mistral with the prompt tuning method achieved the highest Macro-F1 score in both validation and test sets across all sub-tasks, hence, revealing the potential of a causal language model like Mistral to perform sequence classification with the appropriate prompt. For this reason, the predicted test set labels of the prompt tuning Mistral models were selected as the final submissions and received a rank based on their results.

[5] https://pypi.org/project/emoji/
[6] https://pypi.org/project/wordsegment/
[7] https://github.com/cbaziotis/ekphrasis

Validation Set	
Sub-task A	
Model	**Macro-F1**
Mistral LoRA	0.7942
Mistral Prompt Tuning	0.8385
Model	**Macro-F1**
Sub-task B	
Model	**Macro-F1**
Mistral LoRA	0.5829
Mistral Prompt Tuning	0.6071
Sub-task C	
Model	**Macro-F1**
Mistral LoRA	0.5854
Mistral Prompt Tuning	0.6446
Test Set	
Sub-task A	
Model	**Macro-F1**
Mistral LoRA	0.7990
Mistral Prompt Tuning	0.8649
Sub-task B	
Model	**Macro-F1**
Mistral LoRA	0.5713
Mistral Prompt Tuning	0.6106
Sub-task C	
Model	**Macro-F1**
Mistral LoRA	0.6160
Mistral Prompt Tuning	0.6930

Table 3: Results of all models on test and validation sets based on Macro-F1 score.

Specifically, in sub-task A, the Mistral prompt tuning method achieved the 10^{th} place out of 22 submissions with a Macro-F1 score of 0.8649. In subtask B, it achieved the 11^{th} place out of 18 submissions with a Macro-F1 score of 0.6106. Lastly, in sub-task C, it achieved the 13^{th} place out of 19 submissions with a Macro-F1 score of 0.6930. According to Table 4, it is revealed that the submitted Mistral prompt tuning models managed to beat the baseline accuracy and Macro-F1 scores of the models developed by the dataset creators across all sub-tasks (Shiwakoti et al., 2024). The dataset creators have experimented with Transformer models like BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2020), RoBERTa (Liu et al., 2019) and ClimateBERT (Webersinke et al., 2021) using a batch size of 16 for 3 epochs with a learning rate of 1e-5 for DistilBERT and 1e-3 for the rest of the models. By taking into account the Macro-F1 score of each class on the validation and test sets in Table 5, it is demonstrated that the models are not able to identify the *COMMUNITY* minority class and, surprisingly, the *NEUTRAL* majority class, since they achieved the lowest scores. On the other hand, the *NON-HATE* and *INDIVIDUAL* majority classes yielded the highest scores. In subtask A, both models can identify non-hateful content more accurately than hateful content. However, the Mistral Prompt Tuning model outperforms the Mistral LoRA model in detecting hateful tweets. In subtask B, the models successfully detect individuals as targets of hate speech, but fail to identify organizations and communities. Both models in sub-task C perform better at identifying stances that show support or opposition rather than neutral stances. The Mistral Prompt Tuning model exhibited better performance in the support and oppose classes compared to the Mistral LoRA model. The Mistral LoRA model's performance was higher in identifying the *OPPOSE* stance on the test set than on the validation set, the same applied to the *NEUTRAL* stance as well. Finally, the Mistral Prompt Tuning model achieved a higher score for the *OPPOSE* stance on the test set than on the validation set.

Sub-task A		
Model	**Macro-F1**	**Accuracy**
BERT	0.708	0.901
DistilBERT	0.664	0.896
RoBERTa	0.662	0.842
ClimateBERT	0.704	0.884
Mistral Prompt Tuning	**0.864**	**0.946**
Sub-task B		
Model	**Macro-F1**	**Accuracy**
BERT	0.554	0.641
DistilBERT	0.550	0.603
RoBERTa	0.501	0.716
ClimateBERT	0.549	0.604
Mistral Prompt Tuning	**0.610**	**0.840**
Sub-task C		
Model	**Macro-F1**	**Accuracy**
BERT	0.466	0.586
DistilBERT	0.527	0.610
RoBERTa	0.542	0.648
ClimateBERT	0.545	0.651
Mistral Prompt Tuning	**0.693**	**0.665**

Table 4: Comparison of submitted fine-tuned models with baseline models on test set based on Macro-F1 score and accuracy.

Class Label	Macro-F1 Validation	Macro-F1 Test
Sub-task A		
Mistral LoRA		
NON-HATE	0.9514	0.9478
HATE	0.6371	0.6502
Mistral Prompt Tuning		
NON-HATE	0.9664	0.9697
HATE	0.7107	0.7600
Sub-task B		
Mistral LoRA		
INDIVIDUAL	0.9101	0.9487
ORGANIZATION	0.5660	0.5652
COMMUNITY	0.2727	0.2000
Mistral Prompt Tuning		
INDIVIDUAL	0.9167	0.9487
ORGANIZATION	0.5714	0.5128
COMMUNITY	0.3333	0.3704
Sub-task C		
Mistral LoRA		
SUPPORT	0.6737	0.6835
OPPOSE	0.6169	0.6838
NEUTRAL	0.4657	0.4806
Mistral Prompt Tuning		
SUPPORT	0.7038	0.7195
OPPOSE	0.7250	0.8244
NEUTRAL	0.5052	0.5351

Table 5: Macro-F1 scores in each class on test and validation sets.

5.1 Error Analysis

The confusion matrices were generated after the release of the test set labels. The purpose was to reveal the errors and strengths of the submitted Mistral Prompt Tuning models. Figure 1 displays the performance of the Prompt Tuning models on the test set of sub-tasks A, B and C respectively, through the confusion matrices. Firstly, it is evident from the confusion matrix of sub-task A that the model performed better in identifying tweets that do not contain hate speech. This could be attributed to the limited data available in the *HATE* class. The model placed greater emphasis on boosting the *NON-HATE* class, which further skewed the models' ability to accurately detect hate speech tweets. Moreover, from the confusion matrix of sub-task B, it is evident that the model managed to detect tweets that target individuals with greater confidence and success because it was the majority class. The *COMMUNITY* class contained the least examples in the training set, hence the model was able to classify fewer examples than expected into this category and more examples into the other categories. The model also seemed to have gotten a bit confused when it came to identifying between the *ORGANIZATION* and *COMMUNITY* classes, as texts that belonged to the *COMMUNITY* class were assigned to the *ORGANIZATION* class. Finally, it has been proven that the model found it difficult to distinguish between tweets that belonged to the *SUPPORT* and *NEUTRAL* stance classes in sub-task C, since many texts were falsely classified as expressing support or neutral stance, respectively.

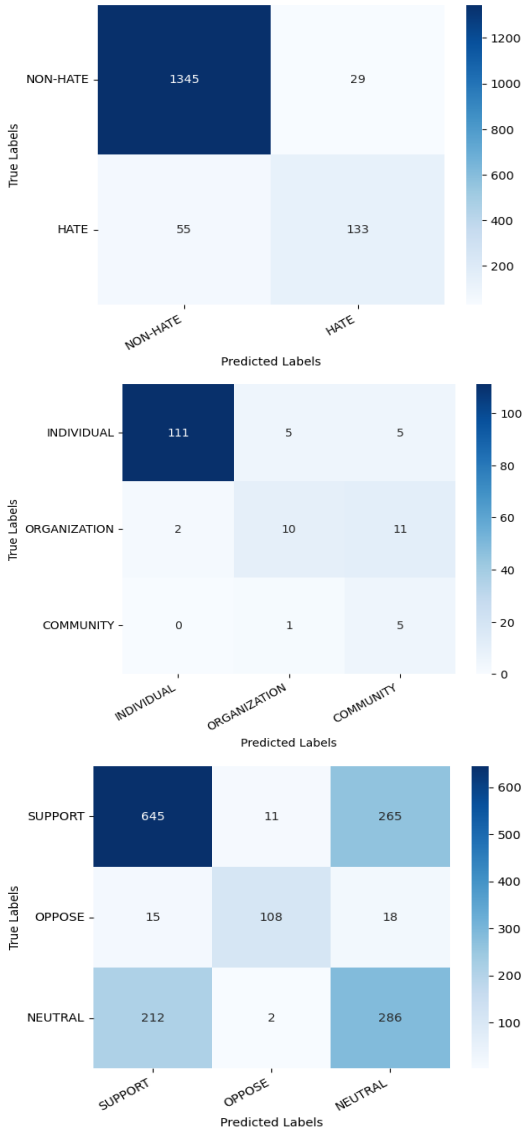

Figure 1: Test Set Confusion Matrices of Mistral Prompt Tuning models.

6 Conclusion

The Climate Activism Stance and Hate Event Detection Shared Task at CASE 2024 involved fine-tuning the LLM Mistral-7B with two PEFT methods (LoRA and prompt tuning) for binary and multi-class text classification. This resulted in the creation of six models that can detect hate speech, targets of hate speech, and stance regarding climate change and activist events. The approach also included adding weights to deal with class imbalance, as well as data cleaning and pre-processing. Comparing the two PEFT methods showed that the prompt tuning method yielded the best performance by crafting the most appropriate and precise prompt for each task. Both methods, particularly the prompt tuning method that was submitted, outperformed all Transformer language models that were fine-tuned by the task organizers and whose scores were presented as baselines. To further improve the models' performance, future efforts should concentrate on adding more tweets in the sub-tasks, especially hate speech and targets of hate speech. Although the Mistral model was originally designed for text generation, it demonstrated its potential to perform sequence classification effectively as well.

7 Limitations

The experimentation process across all sub-tasks revealed a major issue of class imbalance. Despite assigning higher weights to the minority classes, it became clear that detecting hate speech, targets of hate speech, and stances concerning climate change and events was indeed very challenging. The primary reason for this is the scarcity of data available for these categories. The lack of sufficient data causes the trained models to be biased towards the majority classes, which results in poor performance on the minority classes. Unfortunately, there was no other relevant climate activism dataset to leverage for this task. As possible solutions, more data related to climate activism stances and hate events as well as further model experimentation are necessary. More data will certainly help balance the classes and train the models to be less biased and more successful in detecting hate speech, targets of hate speech and stances concerning climate change and events.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Akash Gautam, Puneet Mathur, Rakesh Gosangi, Debanjan Mahata, Ramit Sawhney, and Rajiv Ratn Shah. 2019. metooma: Multi-aspect annotations of tweets related to the metoo movement.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Shuvam Shiwakoti, Surendrabikram Thapa, Kritesh Rauniyar, Akshyat Shah, Aashish Bhandari, and Usman Naseem. 2024. Analyzing the dynamics of climate change discourse on twitter: A new annotated corpus and multi-aspect classification. *Preprint*.

Dominik Stammbach, Nicolas Webersinke, Julia Anna Bingler, Mathias Kraus, and Markus Leippold. 2023. Environmental claim detection.

Kim Taehoon, Tahir Kevin, Wurster, and Jalilov. 2022. Emoji.

Surendrabikram Thapa, Kritesh Rauniyar, Farhan Ahmad Jafri, Shuvam Shiwakoti, Hariram Veeramani, Raghav Jain, Guneet Singh Kohli, Ali Hürriyetoğlu, and Usman Naseem. 2024. Stance and hate event detection in tweets related to climate activism - shared task at case 2024. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*.

Nicolas Webersinke, Mathias Kraus, Julia Anna Bingler, and Markus Leippold. 2021. Climatebert: A pretrained language model for climate-related text. *CoRR*, abs/2110.12010.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

A Appendix

Class Label	Weight
Sub-task A	
NON-HATE (0)	1.1049185563717663
HATE (1)	10.531202435312025
Sub-task B	
INDIVIDUAL (0)	1.4171779141104295
ORGANIZATION (1)	4.4
COMMUNITY (2)	14.903225806451612
Sub-task C	
SUPPORT (0)	1.6295336787564767
OPPOSE (1)	15.106986899563319
NEUTRAL (2)	3.1237020316027087

Table 6: Calculated Weights Based on Class Distribution in Training Sets.

Hyperparams	**Sub-task A**	**Sub-task B**	**Sub-task C**
Classes	2	3	3
Epochs	10	10	10
Seq. Length	195	193	195
Batch Size	16	16	16
Learning Rate	1e-4	1e-4	1e-4
Weight Decay	0.0001	0.0001	0.0001
M. G. Norm	0.3	0.3	0.3
Warm-up R.	0.1	0.1	0.1
AdamW E.	1e-8	1e-8	1e-8
G. A. Steps	2	2	2
Early Stop.	5	5	5
Seed	42	42	42
Virtual Tokens	37	44	45
LoRA r	16	16	16
LoRA alpha	16	16	16
LoRA dropout	0.05	0.05	0.05
LoRA bias	none	none	none

Table 7: Model Hyperparameters in Each Sub-task.