Fine-tuning open-source LLMs for argument detection

Adriana Orellana Angel Zenteno Tim Arni

***Abstract*—The CommPass project focuses on visualizing semantic similarity and polarity in news articles to address filter bubbles while still enabling personalized content consumption. This paper delves into the initial phase of the polarization-identification pipeline: the argument detection task. Leveraging the Low-Rank Adaption (LoRA) technique, three open source Large Language Models (LLMs), namely LLaMa-2-7B, Mistral-7B, and Phi-1.5, are fine-tuned for this task. The models classify sentences as either arguments or non-arguments, contributing to the overall goal of understanding the polarization within news articles. Results show that our best fine-tuned model Mistral-7B achieves an accuracy of 84.24% and F1-Score of 82.23% on the Stab argument detection dataset. We further explore practical applications in analyzing English news articles about the murder of Samuel Paty, extracting arguments with our best model.**

I. Introduction

CommPass is a project by the Distributed Information Systems Laboratory at EPFL, that aims to visualize the semantic similarity of news articles from multiple perspectives by going beyond standard thematic similarity and exploring whether polarity can be visualized in articles covering a given media event. Such visualisation tools can be used to create awareness by media producers and aggregators for the polarity of media contents. This helps users to avoid getting caught in filter bubbles while enabling personalization of content. [1] Partitioning articles for a given news event into fine-grained sub-topics and using embedding models to group together news article with the same semantical meaning works already good, but it is not easy to identify the polarization inside each group yet. The aim of several ML4Science projects is to test, if analysing the argumentative structure of the news article helps to identify the polarization. To do so the pipeline is the following: 1) argument detection, 2) argument stance classification, 3) argument summarisation.

This report focuses on the first step of the polarization-identification pipeline: the argument detection. Using Low-Rank Adaption (LoRA) three different large language models (LLMs) are fine-tuned for argument detection. The fine-tuned models classify a given sentence as an *argument* or *non-argument* by providing a yes or no output, respectively.

[1]https://www.media-initiative.ch/project/commpass/

II. Background

A. Large Language Model

Large Language Models are deep learning models that have been pre-trained on massive datasets of text. They employ transformers [1] which are neural networks featuring self-attention mechanism that utilize embeddings to represent input data effectively. This architecture enables them to generate and understand human language in general-purpose manner, abilities that are essential in tasks such as detecting arguments within sentences. To harness these capabilities, we fine-tune three open-source models: LLaMa 2 7B[2], Mistral 7B[3] and Phi 1.5B[4]. For LLaMa and Mistral we used the versions with 7 billion parameters, while phi-1.5 has 1.3 billion parameters.

B. Finetuning using LoRA

Fine-tuning refers to the process of updating the weights W_0 of the pre-trained model, using data for a specific task to obtain a better performance. ΔW is the weight update and more specifically it is the accumulated gradient during adaption. After fine-tuning the pre-trained weights W are updated with ΔW and the weights of the fine-tuned model are $W = W_0 + \Delta W$. To update all layers can be computationally very expensive, since LLMs have a lot of parameters. To address this, parameter-efficient methods, such as prefix tuning, adapters or LoRA have been developed to significantly reduce the number of parameters that are adjusted and therefore make fine-tuning computationally less expensive. In this study we use Low-Rank adaptation (LoRA) which was introduced by Hu et al. [2] in 2021. The idea is to decompose the weight updates ΔW into lower-rank representations $\Delta W = BA$. When $W_0 \in R^{d\times k}$ and $\Delta W \in R^{d\times k}$, then B $\in R^{d\times r}$ and $A \in R^{r\times k}$, where r is the rank of LoRA and can be much smaller than d and

[2]https://huggingface.co/meta-llama/Llama-2-7b

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

[4]https://huggingface.co/microsoft/phi-1_5

k. If the hyperparameter r is small, less weights have to be learned during fine-tuning, which speeds up the fine-tuning process and makes it computationally much less expensive. During training, W_0 is frozen and only the parameters contained in B and A are trained. The modified forward pass of the fine-tuned model is now $h = Wx = W_0x + BAx$.

III. Methodology

In this section, we will introduce the datasets used for fine-tuning the pre-trained models for argument sentence detection. Additionally, we will detail the data preprocessing steps and the selection of hyperparameters for fine-tuning LLaMa-2-7B, Mistral-7B and Phi-1.5 models.

A. Datasets

1) Stab Dataset: The Stab dataset [3] consists of ∼25k annotated sentence covering 8 controversial topics. The dataset is balanced and consists of ∼14k *non-arguments* and ∼11k *arguments*. To asses the annotation quality and to create the annotation protocol, a group of expert annotators generated a gold standard on a subset of randomly selected sentences. The sentences were then annotated by seven anonymous American workers from Amazon Mechanical Turk each. The labels in the Stab dataset are *non-argument*, *supporting argument* and *opposing argument*, some examples are shown in Table I. To request access to this dataset, follow the instructions here[5]. We used the same partition as proposed by Stab et al. for the training (70% of the samples), validation (10%) and test (20%) sets.

2) IBM Dataset: The IBM dataset[6] contains 700 sentences that were extracted from debate speeches over controversial topics. Each sentence was annotated by three expert annotators and their majority vote was taken as the label. [4] Some examples are shown in Table I. To maintain consistency with the Stab dataset, we divided the IBM dataset into three sets: 16% of the 601 sentences labeled as "test" were utilized as validation set, the remaining percentage as training set, and the 99 sentences labeled as "val" were reserved as test set.

B. Data preparation

1) Data pre-processing: Both, the Stab and IBM dataset provide information about the sentences along with their respective labels. However, there are differences in the labeling approach. In the case of the Stab dataset the labels are "NoArgument", "Argument_against" and "Argument_for". We map "NoArgument" to the "no" label for non-arguments, and all others to "yes", since we are not interested in the stance. Subsequently, we split the dataset in training, testing and validation according to which set the data belongs. On the other hand, in the IBM dataset, a sentence is labeled as "0" for non-arguments and "1" for arguments. We map "0" to "no" for non-arguments, and "1" to yes for arguments.

2) Alpaca style: The data is converted into *Alpaca style*, to be used with the framework proposed by lit-gpt[7] for fine-tuning. Each sentence and its annotation is transformed into a JSON object, containing the same prompt as instruction, the sentence as input and the annotation as output.

IV. Results and Discussion

A. Quantitative Evaluation

In this section, we introduce the baseline models that are used as reference points and show the results obtained with the fine-tuned models.

1) Zero-shot evaluation: Zero-shot prompting involves evaluating a model's performance on tasks for which it has not been explicitly trained on, by providing natural language prompts during testing. We evaluate Llama-2-7b-chat, Mistral-7B-Instruct-v0.1 and GPT-3.5-turbo. Our aim is to assess whether detecting an argument is a task that can be performed by a instruction fine-tuned LLM. Since there is no version of Phi-1.5 that is fine-tuned to follow instructions provided by Microsoft, no zero-shot evaluation using Phi-1.5 is done.

To facilitate this assessment, we provide these models with a prompt that defines the role of the chat model as an argument sentence detector, outlines the definition of an argument, and specifies the expected response format (yes or no) for each input. The whole message was first formatted to the instruction format of each model to ensure accurate and coherent responses. The results are shown in Table II. We note that GPT-3.5-turbo obtains the best accuracy followed by Mistral-7B-Instruct, this preliminary results indicate that it is possible to increase the performance of these models by performing fine-tuning on the argument detection task.

[5]https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2345

[6]https://research.ibm.com/haifa/dept/vst/debating_data.shtml#Argument_Detection

[7]https://github.com/Lightning-AI/lit-gpt

Dataset	Topic	Sentence	Label
Stab	nuclear energy	Nuclear fission is the process that is used in nuclear reactors to produce high amount of energy using element called uranium.	NoArgument
Stab	nuclear energy	It has been determined that the amount of greenhouse gases have decreased by almost half because of the prevalence in the utilization of nuclear power.	Argument_against
IBM	ban anonymous posts	People are always going to find a way to post anonymously	ARGUMENT
IBM	legalize cannabis	They had a vote and people voted to prohibit the sale of alcohol	NON-ARGUMENT

TABLE I: Example sentences from the Stab and IBM datasets

Model name	**Accuracy**	**F1 score**
LLaMA-2-7b-chat	0.5486	0.6465
Mistral-7B-Instruct	0.6555	0.6188
GPT-3.5-turbo	0.7514	0.7574

TABLE II: Zero-shot evaluation results on the Stab test set of several LLMs for argument detection.

2) Hyperparameter Tuning: In this section, we present the hyperparameter tuning process conducted for the Mistral 7B model. Phi-1.5 and LLaMa-2-7b behaved qualitatively similar, but had worse validation losses. The primary focus of our investigation centered on hyperparameters associated with LoRA, specifically the parameters 'r' and 'alpha.' To explore the impact of different LoRA hyperparameter configurations on model performance, we conducted experiments with alternative settings for 'r' and 'alpha'. The following configurations were tested and the validation loss is shown in Figure 1:

- r=8, alpha=16, without masked input
- r=8, alpha=16, masked input
- r=8, alpha=16, masked input, topic in prompt
- r=8, alpha=16, masked input
- r=16, alpha=32, masked input
- r=32, alpha=64, masked input

Increasing the value of 'r' in LoRA was observed to increase training time, what is expected since the number of trainable parameters increases. Results indicated that while increasing these hyperparameters led to faster convergence, the resulting validation loss remained constant across different configurations. Because one training run is very time and computation intensive, we did not perform extensive testing for all models and kept a hyperparameter fixed when a good value was identified during the fine-tuning.

To further understand the model's response to different prompts, we experimented with adding the topic of the argument to the input prompt. The model's ability to discern arguments, measured by validation loss, exhibit a small improvement with this prompt modification.

We also explored the impact of masking inputs during training. This technique involves limiting the model's focus to learn only the output classification ('yes' or 'no'). Masking inputs resulted in a remarkable decrease in validation loss, reducing it 2.8 times after 3000 iterations. The validation loss dropped from 0.51 to 0.18, indicating a significant enhancement. Among the explored hyperparameter configurations, input masking was the most impactful modification, substantially improving the model's performance by simplifying the learning task to binary classification.

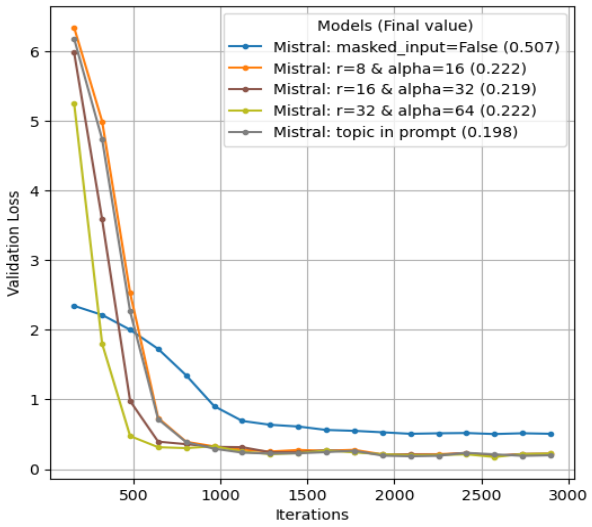

Fig. 1: Hyperparameter finetuning: The figure shows the curve of the validation loss for different combinations of hyperparameter.

3) RoBERTArg:: As a baseline for comparison, we also report RoBERTArg[8] which is a fine-tuned version of the RoBERTa (base) model[9] for argument detection and it was trained on the Stab 2018 dataset. The model was evaluated using the Stab test set and the results are shown in Table III.

4) LoRA fine-tuning evaluation: We fine-tuned Phi-1.5, Llama-2-7B and Mistral-7B using LoRA with one

[8]https://huggingface.co/chkla/roberta-argument

[9]https://huggingface.co/roberta-base

Model/Train set	Stab		IBM		Stab + IBM	
	Accuracy	**F1**	**Accuracy**	**F1**	**Accuracy**	**F1**
Phi	0.6406	0.4724	0.5590	0.0061	0.6106	0.3314
	0.5454	0.4827	0.6363	0.0512	0.5555	0.2903
Llama	0.6284	0.6964	0.6063	0.3709	0.7324	0.7340
	0.5555	0.5600	0.6363	0.5263	0.6363	0.2173
Mistral	0.8109	0.8114	0.7572	0.7197	**0.8424**	**0.8238**
	0.5858	0.5858	0.6464	0.6315	0.6363	0.5263
RoBERTArg	0.8297	0.8195				

TABLE III: Evaluation of LoRA fine-tuning: The grey cells show results on the Stab test set and the white cells on the IBM test set.

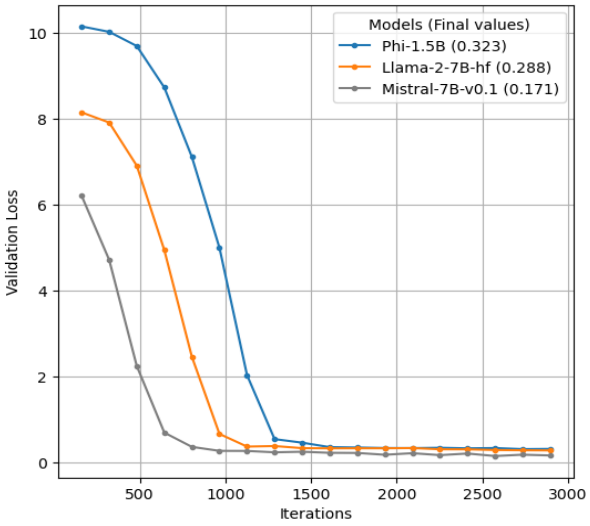

Fig. 2: Fine-tuning validation loss: The figure shows the validation loss for the three models during training with the best combination of hyperparameters that were tested. As shown in III Mistral-7B performs best.

Nvidia A100 GPU with 40 GB of VRAM (see Figure 2). The three models were trained on 3 different training sets (Stab, IBM, Stab + IBM combination) and then evaluated using two different test sets (Stab and IBM). The obtained results including RoBERTArg as baseline are shown in Table III (results on the Stab test set are gray, those on the IBM test set white). After every 160 iteration (approximately one epoch) the validation loss is calculated during training. The best accuracy and F1 score was achieved with Mistral-7B fine-tuned on the Stab + IBM dataset. Thus, this is the selected model to perform the argument extraction.

B. Qualitative Evaluation: Analyzing English news articles about the murder of Samuel Paty

To compare the performance of the best fine-tuned Mistral model to the baseline RoBERTArg on classifiying sentences extracted from newspapers, we use Spacy's sentence segmentation model to split the articles into sentences. The 5027 sentences extracted are then classified as *argument* or *non-argument* by RoBERTArg and fine-tuned Mistral. RoBERTArg classifies 903 sentences as *argument*, while fine-tuned Mistral classifies 242 sentences as *argument*. To compare the quality we take 30 randomly sampeled sentences (15 *argument* and 15 *non-argument*) from the output of both models each. A human annotator classifies each sentence as *argument* or *non-argument*. The human annotator only sees the sentence and a unique hash-id per sentence (used to keep track of the results) and does neither know from which model a sentence is, nor sees the classification done by the model. The human used the explanation of Stab et al. [3] to decide if a sentence is an argument or not. The obtained results are: **accuracy of 0.8333 and F1 score of 0.8387 for fine-tuned Mistral** and **accuracy of 0.6667 and F1 score of 0.6429 for RoBERTArg**. Those results indicate, that the fine-tuned Mistral significantly outperforms RoBERTArg on a sample that is different in structure from the training set. To obtain trustworthy results, the above mentioned approach should be performed with more sentences and several human annotators. Due to time and monetary restriction, we could not perform this in depth analysis.

V. Conclusion

In this report we tackle the argument detection problem identified by the CommPass project. We tested three Large Language Models (LLMs) - LLaMa-2-7B, Mistral-7B, and Phi-1.5 - using the Low-Rank Adaption (LoRA) technique to identify arguments within news articles. The fine-tuned Mistral-7B model was the most effective model with an 84.24% accuracy and an 82.23% F1-Score. This model was then used to analyze and understand news articles, exemplified by its application in studying English articles about the murder of Samuel Paty.

VI. Ethical considerations

In today's democratic societies, the rise of fake news and media polarization presents significant challenges. With the advent of generative language models, the production of fake news could escalate rapidly. The proposed CommPass solution aims to assist media users in discerning whether or not they are caught in a filter bubble. In offering a potentially neutral technical tool, this could reach users, that only consult very biased news and therefore increase the likelihood of them consulting other media sources.

One ethical risk is to wrongfully classify (respectively locate in the media space) legitimate news similar to fake ones. The ethical risk of misclassification by large language models in media content analysis impacts various stakeholders, particularly the general public and media producers. Because it could further undermine public trust in media and providing malicious actors, like authoritarian governments or fake news creators, a tool to silence genuine news or legitimize false narratives, by having a seemingly neutral technological tool classifying legitimate and fake news similarly. The severity of this risk is moderate, given that trust in media depends on many actors and its likelihood is relatively low due to the limited use of these models by bad actors. Specifically in argument detection, the model classifies an argument based on its structure rather than its content, and therefore cannot differentiate between factual and fabricated arguments. [10]

We evaluated this risk through research on fake news [5] and a discussion with Cécile Hardebolle. Despite the technical inability to include resource and fact-checking, we've acknowledged this ethical risk in our project. To mitigate potential misuse, our report explicitly cautions future users about these risks, especially if the models are deployed more broadly. We emphasize that classifying an argument does not determine its factualness or validity, urging users to consider these limitations.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

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

[3] Christian Stab, Tristan Miller, and Iryna Gurevych. Cross-topic argument mining from heterogeneous sources using attention-based neural networks, 2018.

[4] Eyal Shnarch, Leshem Choshen, Guy Moshkowich, Ranit Aharonov, and Noam Slonim. Unsupervised expressive rules provide explainability and assist human experts grasping new domains. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2678–2697, Online, November 2020. Association for Computational Linguistics.

[5] Merten Reglitz. Fake news and democracy. *Journal of Ethics and Social Philosophy*, 22(2):162–187, 2022.

[10] https://skepticalscience.com/history-FLICC-5-techniques-science-denial.html

Appendix A
Fine-tuned model evaluation

To select the best fine-tuned model, we have done the following evaluations:

- Fine-tuning and evaluating phi-1.5 with LoRA on Stab dataset.
- Fine-tuning and evaluating Llama-2-7b with LoRA on Stab dataset.
- Fine-tuning and evaluating mistral-7b with LoRA on Stab dataset.
- Fine-tuning and evaluating on IBM dataset.
- Fine-tuning on the combined dataset (Stab+IBM), evaluating each test set independently.
  Table IV shows the results of this evaluation.

Appendix B
Prompts

The prompts utilized in this project not only define what text qualifies as an argument but also specify the expected format of the answer ('yes' or 'no') to the model. Additionally, the zero-shot system prompt has a sentence to explain the desired behavior that the model should show. Here are the prompts that were employed throughout our project.

- **Fine-tuned models:** Given an input text, classify it as either an argument ('yes') or not an argument ('no'). The input is considered an argument if it either supports or opposes a topic and includes a relevant reason for supporting or opposing the topic. Conversely, if the input does not include reasons, classify it as a non-argument.
- **Prompt for Llama Chat 2-7B and Mistral-7B-Instruct-v0.1:** You are an Argument Sentence Detector. Your task is to identify whether a given sentence is an argument or not. An argument is defined as a set of statements supporting or opposing a claim, exhibiting logical structure with premises and a conclusion. If the text demonstrates this logical structure, the output should be 'yes', while 'no' is appropriate if the text mainly contains claims or verifiable information without a clear

Model	Train	Eval	Accuracy	Precision	Recall	F1-Score
phi	stab	stab	0.6406	0.6716	0.3644	0.4724
llama	stab	stab	0.6285	0.5448	0.9650	0.6964
mistral	stab	stab	0.8109	0.7251	0.9211	0.8114
phi	stab	ibm	0.5455	0.4118	0.5833	0.4828
llama	stab	ibm	0.5556	0.4375	0.7778	0.5600
mistral	stab	ibm	0.5859	0.4603	0.8056	0.5859
phi	ibm	ibm	0.6364	0.5000	0.0270	0.0513
llama	ibm	ibm	0.6364	0.5000	0.5556	0.5263
mistral	ibm	ibm	0.6465	0.5085	0.8333	0.6316
phi	ibm	stab	0.5590	0.6364	0.0031	0.0062
llama	ibm	stab	0.6064	0.6302	0.2629	0.3710
mistral	ibm	stab	0.7573	0.7343	0.7057	0.7197
phi	stab-ibm	stab	0.6107	0.6857	0.2185	0.3314
llama	stab-ibm	stab	0.7324	0.6541	0.8364	0.7341
mistral	stab-ibm	stab	0.8424	0.8137	0.8342	0.8238
phi	stab-ibm	ibm	0.5556	0.3462	0.2500	0.2903
llama	stab-ibm	ibm	0.6364	0.5000	0.1389	0.2174
mistral	stab-ibm	ibm	0.6364	0.5000	0.5556	0.5263

TABLE IV: Results of fine-tuned models evaluation

logical structure. Your response should be limited to a simple 'yes' or 'no' without any additional explanation.