11-667 Course Project: LLM Cocktail

Krish Rana* **Lakshay Sethi*** **Onkar Thorat*** **Pratik Mandlecha***
{krana, lsethi, othorat, pmandlec}@andrew.cmu.edu

1 Introduction

The field of AI has witnessed remarkable progress, notably with the advent of large language models (LLMs) that have revolutionized natural language understanding. Among the various applications, instruction-following tasks have gained substantial attention for their wide-ranging utility in diverse domains. However, as these models grow in size and complexity, so does the computational overhead for training and inference. Ensemble methods like LLM Blender (Jiang et al., 2023) have shown that performance can be enhanced by leveraging the strengths of multiple fine-tuned LLMs. Although LLM Blender's ensembling technique shows significant improvements in accuracy, the computational cost associated with inference remains high creating a barrier to the widespread adoption of LLMs.

The primary objective of this research project is to develop an ensemble of fine-tuned LLMs that not only achieves a high degree of accuracy on the MixInstruct benchmark dataset but also reduces inference time substantially. By optimizing the ensemble's architecture and inference strategies, we aim to fulfill the dual goals of performance and computational efficiency.

2 Related Work

Researchers have been exploring ensemble methods extensively in the domain of traditional Machine Learning. Numerous traditional ML researchers have leveraged Bagging and Boosting techniques to achieve remarkable results, especially in small domain datasets for classification and regression tasks. It is widely accepted that when implemented correctly, ensemble methods almost invariably enhance system accuracy. This natural progression extends to the current era of Large Language Models. Anioł et al., as documented in (Aniol et al., 2019), employed an ensemble approach involving three attention-based models, surpassing the performance of the leading Mnemonic Reader model on the SQUAD dataset at that time. More recently, Jiang et al. demonstrated the superiority of ensembling (Jiang et al., 2023) various open-source LLMs over individual models across a diverse range of NLU tasks. In a different vein, Lee et al. adopted a unique strategy by ensembling a set of LoRA adapters (Lee et al., 2023). Their objectives were twofold: firstly, to retain the robust prior knowledge of pretrained LLMs, and secondly, to mitigate the computational demands of the ensemble system.

*Everyone Contributed Equally – Alphabetical order

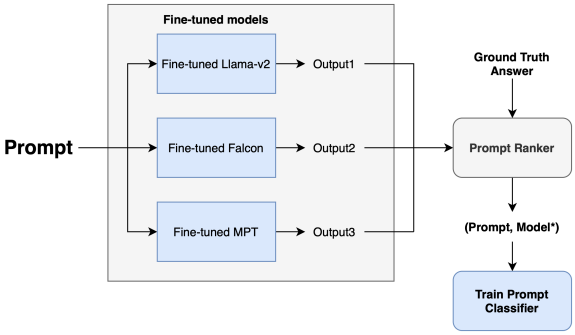

Figure 1: Prompt Classifer Training Pipeline

3 Proposed Methods

The proposed method aims to ensemble language learning model outputs for a given prompt by fine-tuning multiple pre-trained models and employing a prompt ranking and classification system. The method consists of the following key components: Model Fine-Tuning, Prompt Ranking, and Prompt Classification.

3.1 Dataset

We will utilize the MixInstruct dataset (Jiang et al., 2023), which will serve for both fine-tuning our models and training our prompt classifier. To avoid data leakage, we shall use train set for fine-tuning

the models, val set in prompt ranker, and test set for the final evaluation of the entire pipeline.

3.2 Model Fine-Tuning

Three pre-trained and fine-tuned instruct language models: **Llama-v2 7B** (Touvron et al., 2023), **Falcon 7B** (Penedo et al., 2023), and **MPT 7B** (MosaicML-NLP-Team, 2023) will be fine-tuned on the training set of the MixInstruct dataset. To start with, we will use LoRA fine-tuning itself or explore other PEFT methods as well.

3.3 Prompt Ranking

A prompt ranker will be employed to evaluate the performance of each fine-tuned model on each prompt using metrics such as Cosine Similarity, BLEU Score or other complex mechanisms. The ranker will use the validation set to associate each prompt with the model that provides the most accurate output. The prompt ranker will associate each prompt with the corresponding highest ranked model for training the Prompt Classifier.

3.4 Prompt Classification

A prompt classifier will be trained on the associations made by the prompt ranker. For training, it shall receive the prompt and the corresponding model out of the three models under consideration. We shall need to experiment with the kind of model required for this task. In case we need to capture long-range dependencies within the prompt, we may have to use the attention-based models.

3.5 Inference Pipeline

In the inference stage, a prompt will be passed through the trained prompt classifier to identify the most suitable model for generating the response. The prompt will then be processed by the selected model to produce the final response. Our method shall thus not only increase the accuracy but also reduce the inference time as compared to the LLM Blender since we use only one Language Model for each prompt.

4 Proposed Evaluation

To evaluate our ensemble model and its generated text, we intend to employ a comprehensive set of NLG evaluation metrics. Specifically, we will utilize the following established metrics: BERTScore (Zhang et al., 2019), BLEU (Papineni et al., 2002), BARTScore (Yuan et al., 2021), GPT-Rank (Jiang et al., 2023). These selected metrics collectively encompass a diverse array of tasks within the domain of NLG and possess the capacity to accommodate semantically similar terms with shared meanings. It is worth noting that the field of Language Model evaluation is subject to rapid evolution, thus we remain receptive to the possibility of revising the evaluation criteria should an alternative criterion emerge as more pertinent.

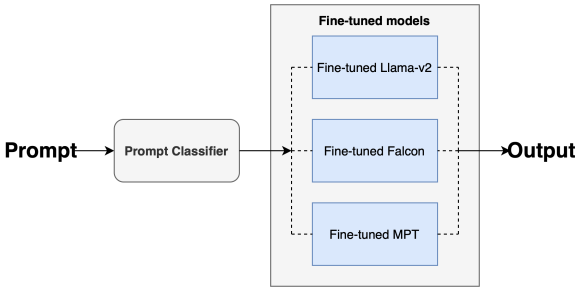

Figure 2: Inference Pipeline

We will be comparing our proposed method with LLM-Blender (Jiang et al., 2023). It is our contention that while the LLM-Blender's approach is robust and achieves commendable accuracy, it is accompanied by a noteworthy computational cost, particularly in terms of FLOPS, compute resources, and inference time. This high computational demand may render it unsuitable for deployment in production systems.

5 Compute

The scope of this project involves fine-tuning large language models and also training a classifier model. The models involve around 7 billion parameters and require an extensive computational power for efficient training. Although we shall use several PEFT methods, we anticipate requiring multiple NVIDIA A100/H100 GPUs that could be available over AWS, GCP or Runpod. We would appreciate it if we could get extra credits for any of these platforms in case the provided $150 on AWS is expected to be used for the Homeworks.

6 Expected Ethical Implications

We are dedicated to addressing significant ethical concerns such as bias, fairness, privacy and informed consent, transparency and explainability, potential harm, accountability, accessibility, inclusivity for diverse groups and responsible deployment throughout our project to maintain the highest ethical standards and benefit society.

7 Team Details

Our team will use Slack for daily communication and Zoom for meetings. We expect each member to contribute to their assigned tasks regularly. We aim for a high-quality research paper and will communicate frequently, with brief catch-ups every couple of days. Our goal is to produce publishable research.

References

Anna Aniol, Marcin Pietron, and Jerzy Duda. 2019. Ensemble approach for natural language question answering problem. In *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)*, pages 180–183. IEEE.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms.

MosaicML-NLP-Team. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *CoRR*, abs/2106.11520.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.