Predicting Human Preferences for LLM responses

SRIHARI INUKURTHI, SULTANA YEAMON, University of Colorado Denver, USA

This research explores the critical challenge of predicting human preferences for outputs generated by large language models (LLMs). Recognizing the growing importance of human-centric AI, this project aims to develop a robust model that accurately reflects user preferences without necessarily enhancing the LLM's underlying capabilities. Inspired by the LMSYS: Chatbot Arena competition on Kaggle, the study employs a novel approach utilizing LLMs fine-tuned with Low-Rank Adaptation (LoRA) and incorporating custom weights. This methodology allows for efficient and effective model adaptation while minimizing the computational cost. By analyzing the intricate relationship between LLM outputs and human expectations, this research seeks to gain a deeper understanding of user preferences and their impact on the LLM's performance. The findings of this study have significant implications for the development of more human-centric and user-friendly AI systems, paving the way for more seamless and intuitive human-computer interactions in various applications.

CCS Concepts: • **Generative AI, Large Language Models, Low-Rank Adaptation**;

Additional Key Words and Phrases: Transformers, Weights, Models

ACM Reference Format:
Srihari Inukurthi, Sultana Yeamon. 2024. Predicting Human Preferences for LLM responses. 1, 1 (December 2024), 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 The Problem

Large Language Models (LLMs) are rapidly becoming an integral part of our daily lives, powering a wide range of applications. However, a critical challenge lies in ensuring that LLM outputs consistently align with human expectations. While LLMs excel at generating text, translating languages, and even composing creative content, their responses often fall short of meeting human preferences. This discrepancy between LLM outputs and human expectations underscores the need for innovative approaches to bridge this gap and ensure more human-centric and satisfying interactions with these powerful AI systems. This project directly addresses this critical challenge by developing and evaluating models that effectively predict human preferences for LLM-generated responses. This discrepancy arises from various factors, including the inherent limitations of current LLM architectures, the lack of robust mechanisms for capturing and incorporating human preferences during model training and generation, and the difficulty in accurately predicting how humans will perceive and evaluate LLM-generated content. This misalignment between LLM outputs and human preferences presents significant challenges for the widespread adoption and effective utilization of these powerful technologies. For instance, in applications such as customer service chatbots, inaccurate or inappropriate responses generated by LLMs can lead to frustrated users, diminished customer satisfaction, and damage to brand reputation. Similarly, in creative

Author's address: Srihari Inukurthi, Sultana Yeamon, srihari.inukurthi@ucdenver.edu,yeamon.sultana@ucdenver.edu, University of Colorado Denver, Denver, Colorado, USA.

ACM XXXX-XXXX/2024/12-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

writing or content generation tasks, LLMs may produce text that is technically proficient but lacks the nuance, originality, or emotional resonance that resonates with human audiences. To bridge this gap, it is crucial to develop effective methods for predicting and aligning LLM outputs with human preferences. This involves understanding the complex interplay between LLM architectures, training data, and human judgment. By developing models that can accurately predict human preferences, we can enable the creation of LLMs that generate outputs that are not only technically sound but also engaging, relevant, and genuinely appreciated by human users.

1.2 Project Objective

This project aims to address this challenge by developing a novel approach that leverage LLMs fine-tuned with Low-Rank Adaptation (LoRA) and incorporates custom weights. This approach focuses on understanding and predicting human preferences without directly enhancing the LLM's core capabilities. By analyzing the alignment between LLM outputs and human expectations, this research seeks to contribute to the development of more human-centric and user-friendly AI systems that can better meet the needs and expectations of human users

1.3 Data

The data sources we intend to use are from the Kaggle dataset[9] repository and the Hugging face dataset repository. These repositories provide good quality data that is required for building a model that can meet the industrial standards provided we have the computational capabilities. These do not have any external requirements. All we need is a Kaggle account and a huggingface account to download the dataset. The dataset contains 9 columns. The training dataset includes over 55,000 real-world user and LLM conversations and user preferences across over 70 state-of-the-art LLMs, such as GPT-4, Claude 2, Llama 2, Gemini, and Mistral models. Each sample represents a battle consisting of 2 LLMs which answer the same question, with a user label of either prefer model A, prefer model B, tie, or tie (both bad). These datasets are carefully curated to ensure diversity and representativeness in various domains and conversational contexts. The characteristics of the data that are worth noting are the "prompt" which includes a wide range of conversational styles, topics, and emotional expressions, "response_a" and "response_b" which exhibit varying levels of quality, fluency, and relevance to the human input, and "winning_model" which represent subjective evaluations of LLM responses, reflecting individual preferences and biases.

2 BACKGROUND

This project helps many stakeholders like AI researchers, Businesses, and customers of LLMs. AI researchers are interested in developing more human-centric and user-friendly AI systems that better understand and respond to human needs and expectations. Companies that utilize LLMs in customer service, content creation, and other applications are eager to improve user satisfaction and enhance the effectiveness of their AI-powered products and services. End-users of LLM-powered applications benefit from systems that generate outputs that are more helpful, informative, and enjoyable. Overall, anyone who interacts with or develops LLMs has a stake in ensuring that these powerful technologies generate output that align with human preferences.

2.1 Potential Impact

2.1.1 Better Alignment. : By understanding and addressing human preferences, LLMs can be fine-tuned to generate outputs that are more human-centric and better suited to specific user needs and contexts.

2.1.2 Improved trust. : When LLMs consistently generate outputs that align with human expectations, it builds trust between users and AI systems.

2.1.3 Enhanced creativity. : LLMs that can effectively capture and reflect human preferences can be powerful tools for creativity and innovation.

2.2 Informal Success Measures

2.2.1 Accuracy. : The primary measure of success will be the accuracy of the model in predicting human preferences, evaluated using metrics such as accuracy, precision, recall, and F1-score.

2.2.2 Efficiency. : The computational efficiency of the model is crucial for practical applications, especially when dealing with large datasets and real-time scenarios.

2.2.3 Interpretability. : The ability to understand and explain the model's predictions is important for building trust and identifying potential areas for improvement.

3 LITERATURE REVIEW

In general, these type of problems/challenges align with the concept of "reward models" or "preference models" in RLHF(Reinforcement Learning from Human Feedback). Previous research has identified limitations in directly prompting an existing LLM for preference predictions. These limitations often stem from biases such as favoring responses presented first (position bias), being overly verbose (verbosity bias), or exhibiting self-promotion (self-enhancement bias). There has been a lot of discussion going on whether LLMs can really understand and respond to the emotional quotient of humans and are really capable of generating responses using the vast information used for training without any bias. There has been some great works which include [2] that helps in integrating the human interaction modalities to the reinforcement learning loop, increasing sample efficiency and enabling real-time reinforcement learning for LLMs. One of the prominent works is the use of reward design by prompting a large language model (LLM) as a proxy reward function, where the user provides a textual prompt using prompt engineering techniques like zero shot or few shot prompting included in [7]. An in-depth analysis of human and LLM preferences[8], based on real-world user-model conversations, uncovered notable discrepancies. Human users demonstrated a higher tolerance for minor errors and a strong preference for responses that aligned with their own perspectives and beliefs. In contrast, advanced LLMs exhibited a stronger emphasis on factual accuracy and clarity, often prioritizing correctness over the potential to reinforce or support user viewpoints. One such similar work is the Preference Proxies[13] explores the potential of LLMs to serve as effective proxies for human preferences in collaboration tasks. The study focuses on explicability and sub-task specification, providing insights into LLMs' ability to model human mental states and reasoning processes. The other approaches include fine tuning LLMs with respect to the scope of the problem that we try to solve and modify the parameters accordingly using the best practices from [11]. The methodologies employed in these include fine-tuning LLMs with human feedback, analyzing real-world user interactions, and developing scenarios where optimal AI performance relies on modeling human mental states and reasoning. These approaches aim to enhance the alignment of LLMs with human preferences and improve their applicability in various contexts.

4 SOLUTION

4.1 Overview

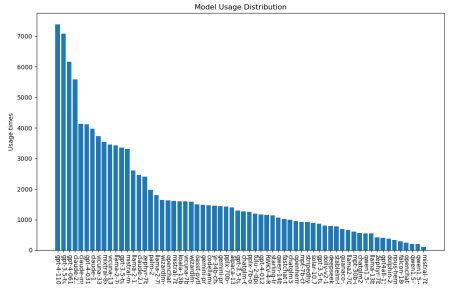
In this project, we explore a novel approach that leverages the inherent structure and relationships within the data itself. We propose a self-supervised learning framework that leverages the inherent

redundancy and consistency within the LLM-generated responses to learn a model that can predict human preferences

4.2 Exploratory Data Analysis

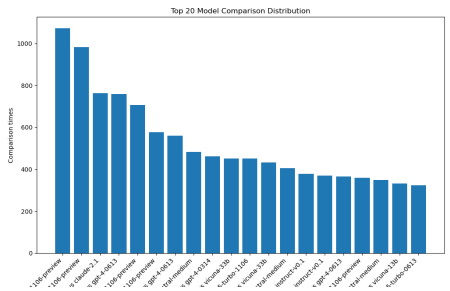
The EDA phase aimed to gain a comprehensive understanding of the dataset, identify key trends, and inform subsequent modeling decisions. The initial inspection revealed the structure and data types of the columns, providing a foundational understanding of the data. To analyze and understand the model usage distribution, a bar chart visualizing the frequency of each LLM model appearing in the dataset. This revealed that certain models were used more frequently than others, suggesting potential biases in the dataset. This information can be crucial for model selection and training, as models with higher usage frequencies may have more data available for training and potentially exhibit better performance. The bar chart can be found at 1

Fig. 1. Model Usage Distribution



.The analysis identified the most frequent model comparisons (e.g., "model_a" vs. "model_b"). This information can help to prioritize model comparisons for further analysis and potentially identify areas where human preferences are more nuanced and challenging to predict. The win rate for each model was calculated, indicating its overall performance across all comparisons. This analysis revealed top-performing and worst-performing models, providing valuable insights into the relative strengths and weaknesses of different LLM architectures and configurations. The presence of significant performance disparities between models suggests that certain models may inherently produce outputs that are more likely to be preferred by humans. The top models can be found at 2 .Further analysis, such as text analysis and sentiment analysis, is crucial to gain a

Fig. 2. Top 20 Models

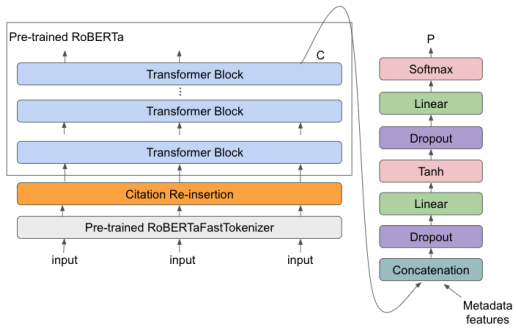


deeper understanding of the factors that drive human preference in the context of LLM-generated responses.

4.3 Methods

4.3.1 Using Roberta base model. The core of the approach revolves around fine-tuning a pre-trained RoBERTa model for the task of human preference prediction motivated from [10] and [1]. This leverages the power of transfer learning, where the model's initial weights, trained on a massive text corpus, are adapted to the specific task of distinguishing between preferred and less preferred LLM responses.The roberta-base model from the Hugging Face Transformers library was chosen as the foundation. This pre-trained model, equipped with a robust transformer architecture, has demonstrated strong performance across various natural language processing tasks. The AutoModelForSequenceClassification class from Transformers was utilized, configuring the model with a single output neuron for binary classification – predicting the probability of model_a being preferred. A crucial component was the creation of a custom BERTDataset class. This class efficiently handled data loading and preprocessing. It tokenized the input text using the AutoTokenizer from Transformers, converting the text into a numerical representation suitable for the model. This process involved encoding the input sequences, adding special tokens (e.g., [CLS], [SEP]), truncating sequences to a maximum length, and padding shorter sequences to ensure consistent input dimensions. The dataset also included the corresponding labels (0 or 1) indicating the preferred model in each pair. Data was loaded and processed into batches using DataLoader from Pytorch for efficient training and evaluation. The model was then trained using the AdamW optimizer with a carefully chosen learning rate and weight decay. A learning rate scheduler was employed to dynamically adjust the learning rate during training, optimizing the learning process. The model architecture can be found at 3 and is inspired from [6]. The training process involved

Fig. 3. Roberta Model Architecture



iterating over multiple epochs, where the model was presented with batches of training data. For each batch, the model generated predictions, the loss was calculated using Mean Squared Error (MSE) between the predicted probabilities and the actual labels, and the model's parameters were updated using backpropagation. CUDA GPU was used to accelerate the training process.

To enhance the model's generalization ability and prevent overfitting, 5-fold cross-validation was implemented. This involved splitting the training data into five folds, training the model on four folds, and evaluating its performance on the held-out fold. This process was repeated five times, with each fold serving as the validation set once. The average performance across all folds provided a robust estimate of the model's true performance. Finally, the best-performing model from the cross-validation process was used to generate predictions on the unseen test data.

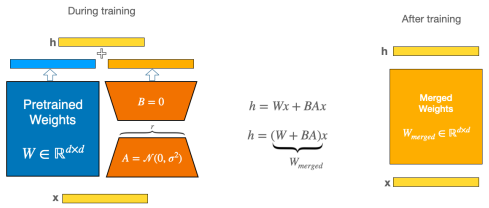
4.3.2 Large Language Models with LoRA and PEFT. This approach relies on adapting one large-scale, pre-trained language model to multiple downstream applications. To fine-tune the models that has huge number of parameters it is difficult with limited computable resources. So, we employ LoRA and PEFT for fine tuning the models to this task. LoRA stands for Low-Rank Adaptation. It is a method used to fine-tune large language models (LLMs) by freezing the weights of the LLM and injecting trainable rank-decomposition matrices. The number of trainable parameters during fine-tuning will decrease therefore considerably. According to [5] and [4], this number decreases 10,000 times, and the computational resources size decreases 3 times. In conventional fine-tuning, weight(W) is updated as

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial L}{\partial \mathbf{W}} = \mathbf{W} + \Delta \mathbf{W} \tag{1}$$

where L is the loss and η is learning rate. LoRA tries to approximate the $\Delta \mathbf{W} \in \mathbb{R}^{d \times k}$ by factorizing $\Delta \mathbf{W}$ into two (much) smaller matrices, $\mathbf{B} \in \mathbb{R}^{d \times r}$ and $\mathbf{A} \in \mathbb{R}^{r \times k}$ with $r \ll \min(d, k)$.

$$\Delta \mathbf{W}_s \approx \mathbf{BA}$$

Fig. 4. LoRA fine-tuning.



Parameter-Efficient Fine-tuning (PEFT) is a technique that optimizes the process of adapting pre-trained language models to specific downstream tasks. Instead of fine-tuning the entire model, which can be computationally expensive, PEFT focuses on adjusting only a small subset of the model's parameters. This approach significantly reduces the number of trainable parameters, leading to faster training times and reduced memory consumption.

PEFT achieves this efficiency by freezing most of the pre-trained model's parameters while allowing a small subset of parameters, often located in specific layers or modules, to be fine-tuned. This targeted fine-tuning allows the model to learn task-specific information while preserving the valuable knowledge encoded in the pre-trained weights.

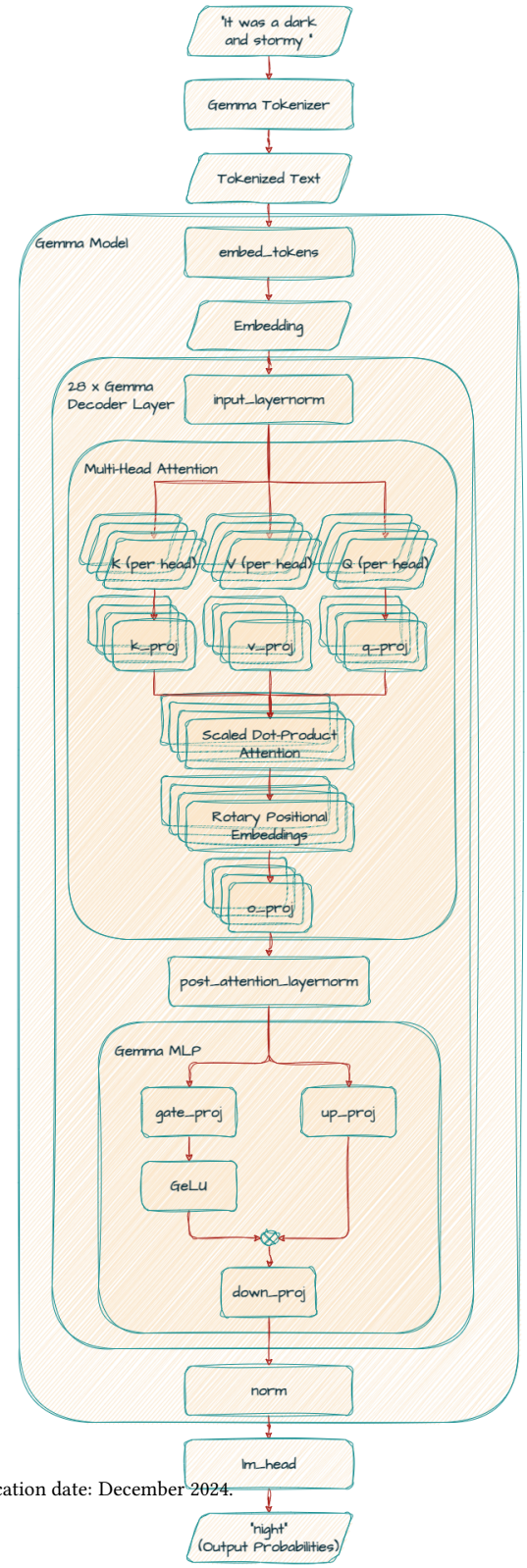
By minimizing the number of trainable parameters, PEFT enables efficient adaptation of large language models to diverse downstream tasks with minimal computational overhead. This makes it a valuable technique for a wide range of applications, including those with limited computational resources.

This approach leverages the power of the Llama 2 8b model[3], fine-tuned with the Lora technique, to predict human preferences for LLM-generated responses. To enhance efficiency, a distributed inference strategy was employed, utilizing two GPUs for concurrent processing. The test dataset was loaded and preprocessed. The prompt, response_a, and response_b were concatenated into a single text string for each example, providing essential context to the model. This concatenated text was then tokenized using the provided tokenizer, ensuring consistent input lengths and padding for efficient processing by the model. Two instances of the Llama 2 8b model were loaded onto separate GPUs, utilizing 8-bit quantization for memory efficiency. The pre-trained Lora weights

were then loaded onto each model. To optimize inference, the models were set to evaluation mode. A key aspect of this approach was the implementation of distributed inference. The input data was divided into two equal halves, with each half assigned to a separate GPU. Two threads were created, each responsible for performing inference on its respective data subset using the corresponding GPU and model instance. This concurrent processing significantly accelerated the inference process. An inference function was defined to efficiently process batches of data on each GPU. This function generated predictions for each class and collected the results. Finally, the predicted probabilities for each class were calculated using softmax.

One more approach is the use of Google's Gemma2[12].

Fig. 5. Gemma Model Architecture.



It uses the same input as the input provided to the Llama model above. The only difference is in the configuration settings. This model uses different tokenizers, LoRA checkpoints, uses the spread_max_length feature to distribute the maximum length across the prompt and responses for more efficient tokenization. This approach loads the model with rank 16 and max length as 2048 instead of rank 16 and batch size as 1024 for the Llama model to allow it to capture more information and improve performance. Also, it performs test-time augmentation by swapping response_a and response_b and repeating the inference process. This approach leverages 8-bit quantization, distributed inference, and test-time augmentation for improved performance and speed.

The LoRA mechanism updates the output projection and value projection modules for the Llama model but the same mechanism with respective configuration updates the query projection and the key projection modules to compute the attention scores between the query and key vectors to improve the performance of the baseline LLM. The LoRa_alpha used with respect to the Gemma model is 32 whereas it is 8 in Llama model approach. It is a scaling factor that controls the degree to which the low-rank matrices (B and A) affect the output during the forward pass. The higher value increases the influence of the low-rank matrices on the output, which speeds up convergence during training.

5 RESULTS AND DISCUSSIONS

At first, we have employed different pre-trained models like Roberta, Llama, Gemma, to test which suits better for the task. Although it not only depends on model that we are using but also the pre-trained weights that we use for the downstream task. The models that are used are possible smaller versions with less parameters. This is because of the computational resource constraint required to train and test these models. For the purpose of evaluation, we test our method on the samples that were not seen by the model during training. The model is tested using the Kaggle's submission mechanism where we submit the Jupyter notebook and it is executed and evaluated against the unseen data. The evaluation metric that is used here is the cross-entropy(log) loss between the predicted probabilities and the ground truth values.

$$L_{\log}(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) \tag{2}$$

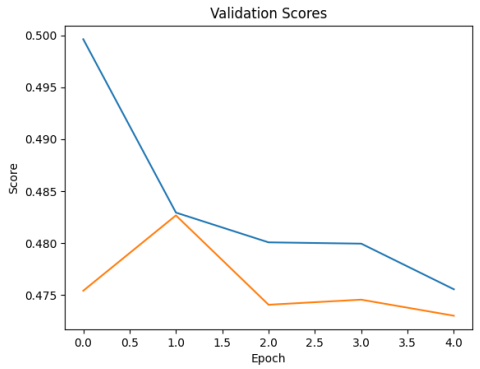
The Gemma model consistently demonstrated superior performance over the other approaches achieving the lowest error score across all experiments. This highlights the model's ability to effectively capture subtle nuances in language and effectively predict human preferences. Notably, the model achieved high accuracy levels with significantly fewer training epochs compared to larger models, resulting in substantial time and resource savings. Beyond quantitative metrics, qualitative analysis of the model's predictions provided valuable insights. Examination of incorrect predictions revealed patterns and areas for improvement, such as difficulties in handling complex or nuanced comparisons and sensitivity to specific linguistic features.

The Gemma model has high accuracy with a loss score of 1.02547. The Llama model has accuracy with a loss score of 1.16277. The combination of Llama 2's powerful architecture and the efficiency gains of LoRA resulted in a strong balance between performance and resource utilization. While exhibiting reasonable performance, RoBERTa generally achieved lower accuracy compared to the larger models, with a loss score of 1.53425. The Gemma model's lightweight architecture and optimized inference pipeline resulted in significantly faster training times and lower latency during inference compared to the other models. The Llama showed a significant improvement in training and inference speed compared to the base Llama 2 7B model, demonstrating the effectiveness of LoRA in reducing computational overhead. The changes to different hyperparameter settings(e.g., learning rate, batch size, LoRA rank) and tokenizers has showed great impact on model performance

and training time. Also, experiments with data augmemtation techniques like concatenation, paraphrasing, etc showed marginal improvements in terms of accuracy for some approaches.

The model's evaluation generated results that showed a satisfactory performance and precision in estimating the probabilities for the prompts and generated response texts. The plot that shows the change in loss for Roberta model while training is shown in 6.

Fig. 6. Score Plot



The primary solution is the Gemma model fine-tuned on the dataset and while inference we also use TTA to improve the performance of deep learning models during inference by applying various transformations to the input data and averaging the predictions thereby increasing the robustness of the model. The naive solution is using the Roberta model for this task.

	Roberta	Llama	Gemma
Log loss	1.53425	1.16277	1.02547
Inference time	-	5s	3s
Training time	4m 12s	2h 3m	1h 36m
Compute Device	GPU T4 x 2	TPU VM v3-8	TPU VM v3-8

Table 1. Results Comparison

Table 1 shows the qualitative results of our primary solution with the other naive solutions that were employed. The results show that the gemma model was able to predict the probabilities better than the other approaches.

6 TOOLS

At the very core of this work is the Generative models that are capable of natural language generation and can be fine-tuned to downstream tasks. The primary tool employed for sequence classification was Gemma2, a large-scale transformer model that excels in natural language processing tasks. Given the task of classifying chatbot responses, where the model had to distinguish between a superior response and a tie, Gemma2's pre-trained capabilities were particularly well-suited. Its transformer architecture, built to handle complex linguistic patterns, made it an ideal candidate. We fine-tuned Gemma2 using LoRA (Low-Rank Adaptation), a method that allows the model to

be adapted more efficiently without retraining all of its parameters. For model deployment and parallelization, we utilized PyTorch, a framework known for its flexibility and high-performance capabilities, particularly in handling large-scale deep learning models. PyTorch allowed us to leverage multi-GPU processing, which was essential given the size and complexity of the transformer model. The framework also provided robust memory management, ensuring that the large model could operate effectively within the available resources. Additionally, PyTorch's support for automatic mixed precision enabled faster processing without sacrificing accuracy. The Hugging Face Transformers library was used to facilitate the integration of the pre-trained Gemma2 model. This library is well-established in the NLP community, providing access to a wide array of pre-trained models and tools for fine-tuning. In terms of data processing and evaluation, Pandas was chosen for its powerful data manipulation capabilities. The tool was instrumental in reading and transforming the dataset, which primarily consisted of text pairs to be evaluated by the model. Additionally, PeFT (Parameter-Efficient Fine-Tuning) was employed to fine-tune the pre-trained model with minimal computational overhead. This technique is particularly useful when working with large transformer models, as it reduces the number of parameters that need to be adjusted during fine-tuning. While these tools formed the backbone of the project, we also considered other options that were ultimately not used. For example, TensorFlow was initially considered as an alternative framework for model deployment. However, after assessing the specific needs of the project, including the integration of pre-trained transformers and fine-tuning, we determined that PyTorch offered superior support for these tasks. Another tool that was explored but not ultimately used was Dask, a library for parallel data processing. Although Dask could have been beneficial for scaling data manipulation tasks, we found that PyTorch's built-in parallelization capabilities were sufficient to handle the demands. Thus, the added complexity of integrating Dask into the workflow was not justified.

7 LESSONS LEARNED

This project aimed to assess the effectiveness of the Gemma2 model, fine-tuned with LoRA, in a sequence classification task which was considered and solved as a regression task because of it's flexibility to evaluate model capability. The model was tasked with evaluating paired chatbot responses to prompts and assigning probabilities for the superior response or a tie. Through a robust and computationally efficient implementation, the model demonstrated its capability to handle complex linguistic patterns and provide meaningful insights into response quality. The Gemma2 model, enhanced with domain-specific LoRA checkpoints, performed exceptionally well in differentiating between chatbot responses. The tokenization process, designed with maximum sequence length constraints and dynamic padding, ensured that the model could handle variable-length inputs effectively. This approach minimized computational overhead and allowed the model to process inputs with varying complexity, maintaining consistent accuracy across all test cases. A notable feature of the implementation was the use of multi-GPU parallelization. The model was loaded onto two GPUs to distribute the computational workload, significantly reducing inference time without compromising prediction quality. This approach, combined with a thoughtful batching strategy, ensured that the model could handle large datasets efficiently while leveraging the advantages of dynamic memory allocation. One of the most compelling aspects of this study was the application of Test-Time Augmentation (TTA). TTA involved flipping the order of responses (response_a and response_b) and averaging the probabilities across original and augmented predictions. This technique enhanced the robustness of the model's predictions by mitigating order biases inherent in sequential data processing. Without TTA, the model occasionally showed a preference for one response order, a subtle bias likely influenced by the fine-tuning dataset. The use of TTA revealed these dependencies, underscoring the importance of augmentation in ensuring fair and

unbiased evaluation. Additionally, the model's ability to process inputs sorted by length, coupled with dynamic padding, showcased the benefits of leveraging advanced pre-processing techniques for optimizing performance. Sorting inputs reduced the padding overhead and maximized GPU utilization, further enhancing inference speed. The final outputs were presented as probabilities for each response being superior or for the responses being equally good, providing a nuanced understanding of the model's decision-making process.

8 LIMITATIONS

The main limitation is the availability and the quality of the data that is used for training and fine-tuning. A limited or unbalanced dataset might not have fully captured the nuances of real-world chatbot interactions, potentially impacting the generalization ability of the models. In cases where the dataset lacked sufficient diversity in terms of language, cultural context, or conversational tone, the models could struggle to produce high-quality responses across a wider range of inputs. Due to the large size of models like Gemma and LLaMA, there were frequent storage bottlenecks when working with large-scale data. The memory required for training and processing these models often exceeded the available GPU memory. The computational cost of training these models also constrained the number of experiments that could be conducted within the project's timeline. Also, the data that is trained on may contain potential bias because the output labels are determined using human resources, which is prone to have multiple kind of contexts for a given particular prompt and response.

REFERENCES

[1] Kshetraphal Bohara, Aman Shakya, and Bishal Pande. 2023. *Fine-Tuning of RoBERTa for Document Classification of ArXiv Dataset.* 243–255. https://doi.org/10.1007/978-981-99-0835-6_18

[2] Vinicius G. Goecks. 2020. Human-in-the-Loop Methods for Data-Driven and Reinforcement Learning Systems. arXiv:2008.13221 [cs.LG] https://arxiv.org/abs/2008.13221

[3] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas

Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[4] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. LoRA+: Efficient Low Rank Adaptation of Large Models. arXiv:2402.12354 [cs.LG] https://arxiv.org/abs/2402.12354

[5] 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. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[6] Zihan Huang, Charles Low, Mengqiu Teng, Hongyi Zhang, Daniel Ho, Mark Krass, and Matthias Grabmair. 2021. Context-Aware Legal Citation Recommendation using Deep Learning. https://doi.org/10.48550/arXiv.2106.10776

[7] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. 2023. Reward Design with Language Models. arXiv:2303.00001 [cs.LG] https://arxiv.org/abs/2303.00001

[8] Junlong Li, Fan Zhou, Shichao Sun, Yikai Zhang, Hai Zhao, and Pengfei Liu. 2024. Dissecting Human and LLM Preferences. arXiv:2402.11296 [cs.CL] https://arxiv.org/abs/2402.11296

[9] Wei lin Chiang, Lianmin Zheng, Lisa Dunlap, Joseph E. Gonzalez, Ion Stoica, Paul Mooney, Sohier Dane, Addison Howard, and Nate Keating. 2024. LMSYS - Chatbot Arena Human Preference Predictions. https://kaggle.com/competitions/lmsys-chatbot-arena. Kaggle.

[10] 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. arXiv:1907.11692 [cs.CL] https://arxiv.org/abs/1907.11692

[11] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. 2024. The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities. arXiv:2408.13296 [cs.LG] https://arxiv.org/abs/2408.13296

[12] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving Open Language Models at a Practical Size. arXiv:2408.00118 [cs.CL] https://arxiv.org/abs/2408.00118

[13] Mudit Verma, Siddhant Bhambri, and Subbarao Kambhampati. 2023. Preference Proxies: Evaluating Large Language Models in Capturing Human Preferences in Human-AI Tasks. In *ICML 2023 Workshop The Many Facets of Preference-Based Learning*.

Received 14 December 2024