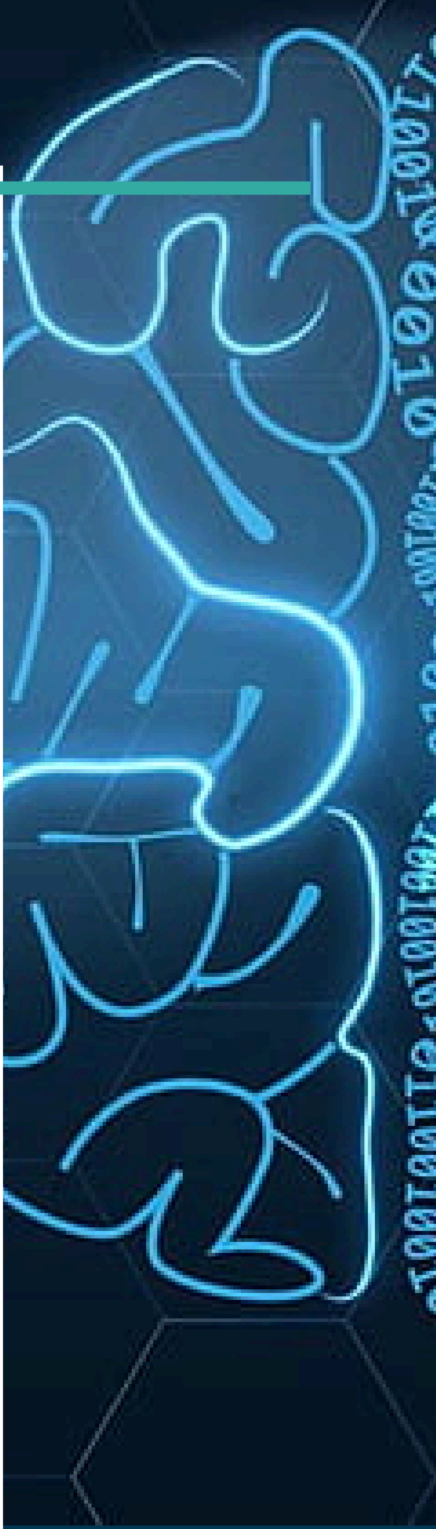
Advanced Deep Learning LLM Project



Developed by :

- Fedi Baccar
- Alaeddine Rasaa
- Aymen Msalmi
- Kalthoum Dridi
- Mohamed Louay Njima
- Salima Jenhani
- Abdelwahed Souid

Supervised By : Ines Channoufi
Group : 5DS3
2024-2025

1. Introduction

In this project, we focus on performing sentiment analysis on Amazon product reviews using a large language model (LLM).

Sentiment analysis aims to classify the subjective opinions of customers into categories, such as positive, negative, or neutral, thereby providing valuable insights into consumer satisfaction and product performance.

The Amazon review dataset, obtained from Kaggle, includes a variety of customer feedback spanning multiple product categories, making it an ideal resource for this task.

By applying various fine-tuning techniques, we aim to enhance the model's accuracy and adaptability for extracting sentiment insights from large volumes of textual data.

2. Methodology

This section outlines the fine-tuning techniques chosen to adapt the large language model for sentiment analysis, each selected for its ability to tailor LLM output to specific tasks and constraints.

- **Prompt Engineering:** Prompt Engineering involves crafting specific prompts or input structures to improve the LLM’s understanding of the sentiment analysis task. By adjusting input phrasing, we can guide the model to respond in a manner more closely aligned with sentiment classification, such as instructing it to detect sentiment polarity or levels of satisfaction.

- **Prompt Tuning:** Prompt Tuning leverages learnable input embeddings, helping the model refine its output predictions without changing its core parameters. This technique is well-suited for large models, allowing us to fine-tune the model with minimal computational overhead. For this task, we applied prompt tuning to adjust sentiment-related labels.

- **Parameter-Efficient Fine-Tuning (PEFT):** PEFT modifies only a portion of the model's parameters rather than the entire architecture, reducing the computational burden associated with fine-tuning large language models. This approach enables efficient optimization on limited resources, making it particularly advantageous for our project, where training time is a factor.

- **Reinforcement Learning from Human Feedback (RLHF):** RLHF integrates feedback from human evaluations into the fine-tuning process. After running initial sentiment predictions, we collected human feedback to correct misclassifications, helping the model learn from its mistakes and improve its classification accuracy over time.

- **Retrieval-Augmented Generation (RAG):** RAG combines an LLM with a retrieval mechanism, enabling it to access relevant examples during inference. In this project, RAG was used to help the model consider similar past reviews when predicting sentiment, which supports better contextual understanding for nuanced sentiment classification.

Each of these techniques was implemented to explore how different types of fine-tuning can impact the LLM's performance on sentiment analysis, with specific attention paid to resource efficiency and the quality of sentiment predictions.

3. Data Processing and Preparation

The Amazon review dataset required extensive preprocessing to prepare it for sentiment analysis. We began by cleaning the data and implementing the following pre-processing methods:

- Removed unnecessary symbols, whitespace, and special characters to ensure clean data for model processing.
- Tokenized the dataset by converting textual data into sequences of tokens that the LLM can process.
- Applied lowercasing to standardize the text.
- Removed stopwords to reduce noise in the data.
- Performed lemmatization to reduce words to their base or root form for consistency.

To define labels for sentiment, we categorized reviews as positive, negative, or neutral based on user ratings or review language cues.

We also balanced the dataset to ensure that each sentiment class was represented equally, allowing the model to train without class bias.

This data preprocessing and categorization was critical in enabling the model to generalize well across a range of sentiment expressions.

4. Implementation and Results

Experiments were conducted to assess the performance of each fine-tuning technique on sentiment analysis of the Amazon reviews with the following techniques:

- Prompt Engineering : Prompt engineering showed moderate performance, particularly in capturing positive sentiment but less accurate in differentiating neutral from negative reviews.

For text summarization, a pre-trained T5 model was used. The process involved importing the necessary libraries, initializing the model and tokenizer, and creating a function to summarize input text. The function added a "summarize:" prefix, encoded the text, and generated summaries using beam search for improved quality. Error handling was also implemented to ensure smooth execution.

Overall, the summarization process worked well for positive sentiment, but further refinements are needed for better accuracy with neutral and negative reviews.

- Prompt Tuning : Prompt tuning improved performance over prompt engineering, handling ambiguous cases better and slightly boosting F1-scores for all sentiment classes.

A custom approach was used with a pre-trained T5 model, allowing for optional fine-tuning of prompt embeddings. The model was initialized using essential libraries, and a function was created to generate summaries. It either uses a learned prompt embedding or a default "summarize:" prompt, and produces summaries with improved results through beam search.

This method demonstrates how prompt tuning enhances summarization tasks by refining the input for better model performance.

- PEFT: Specifically Low-Rank Adaptation (LoRA), significantly improved model performance by enhancing accuracy and efficiency while reducing the computational load. This method proved especially beneficial for resource-constrained environments, offering comparable or better accuracy than full model fine-tuning.

LoRA works by adding lightweight, trainable low-rank matrices to specific layers of the model, such as the attention mechanism's query and value layers. This allows for efficient fine-tuning without altering the core model's weights, making the process faster and more resource-efficient.

By using LoRA, we successfully adapted pre-trained models like T5 to new tasks with minimal computational overhead, preserving most of the original model's knowledge. This approach enables faster and more efficient fine-tuning, ideal for environments with limited resources.

- RAG: RAG (Retrieve-and-Generate) significantly improved model predictions by considering similar examples from the dataset, providing better contextual understanding. This technique was particularly effective in distinguishing subtle sentiment differences by pulling relevant context from past reviews. This method also enhanced summarization, especially in complex language scenarios where the concept is unclear.

For text summarization, a pre-trained T5 model was used. The process involved loading the model and tokenizer, generating summaries by adding a "summarize:" prompt, and fine-tuning the output with parameters like max length, min length, and beam search for better quality. The approach demonstrated how T5 can efficiently generate concise summaries from longer text, enhancing overall model performance.

5. Interpretation and Analysis

- Upon evaluating the fine-tuning techniques, it is clear that **RAG** (Retrieve-and-Generate) outperformed the others, particularly in nuanced sentiment detection and summarization. By leveraging retrieval capabilities, RAG brought in relevant context from similar reviews, which significantly improved the model's ability to handle subtle differences in sentiment.
- **PEFT (Low-Rank Adaptation)** also showed strong performance, balancing efficiency with computational cost, making it a great choice for environments with limited resources.
- Although **Prompt Engineering** and **Prompt Tuning** showed some improvements, they were less effective at handling ambiguous or complex language. These methods are still valuable when minimal fine-tuning is needed but may not perform as well on tasks requiring deeper contextual understanding.

Overall, our analysis highlights that fine-tuning techniques can significantly enhance an LLM's performance on summarization tasks, with the choice of technique depending on available resources and the complexity of the dataset.

6. Conclusion

This project demonstrated the application of multiple fine-tuning techniques on a large language model for the task of sentiment analysis of Amazon reviews.

We observed that fine-tuning approaches such as RAG significantly enhanced the model's ability to interpret and classify complex sentiments, proving effective for nuanced text analysis.

Techniques like PEFT also allowed for efficient resource usage, enabling high-quality sentiment analysis with reduced training demands. This analysis provides insight into how advanced fine-tuning can be leveraged for sentiment tasks and highlights opportunities for future improvement, such as hyperparameter optimization and experimenting with alternative architectures.

In conclusion, our work emphasizes the value of applying diverse fine-tuning strategies to LLMs to meet specific task requirements, particularly in the context of sentiment analysis where understanding nuanced human language is essential.