Enhancing Product Quality Through Sentiment Analysis and Summarization Using LLMs

Scope :

Combining Sentiment Analysis
and Summarization for Product Improvement

Content :

This project aims to enhance product quality by analyzing customer reviews.
The application automatically detects whether customer reviews
are positive or negative. It then separates negative comments
and generates a summary of the issues mentioned, allowing product
teams to make informed decisions to address these issues.

What is Fine-Tuning?
Fine-tuning is the process of taking a pre-trained model and training it further on a specific dataset. This allows the model to adapt to a new, specialized task with improved performance

How It Helped Our Project :

- By fine-tuning the model on our specific dataset of customer reviews (Yelp), we were able to significantly improve its ability to accurately detect sentiment (positive or negative) in the reviews.

- Fine-tuning allowed us to leverage the strengths of a large, general model while customizing it for the specific nuances of customer feedback in our domain.

Dataset Preparation: Data was preprocessed and labeled (Positive/Negative).

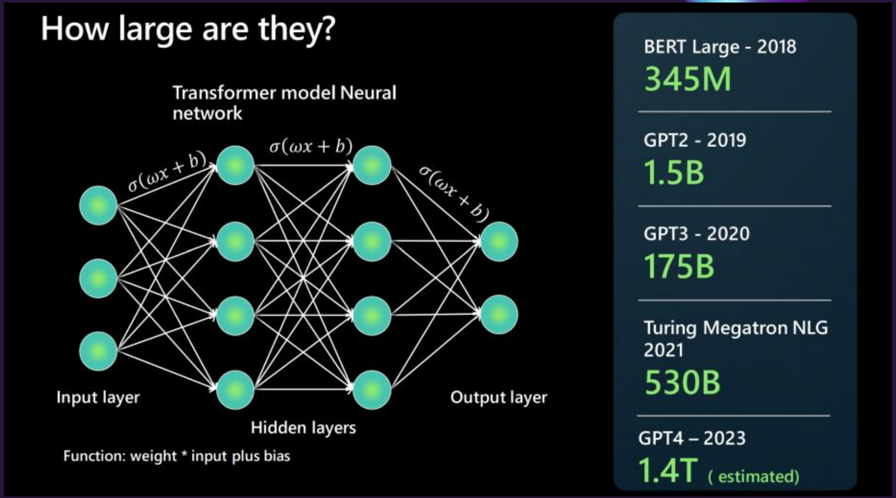
Model Setup:
Used transformer-based models and applied LoRA for efficient fine-tuning

Challenges with Large Models:

Complexity and Resource Demands:

- Pre-trained transformer models are often very large with millions or even billions of parameters

  Fine-tuning all these parameters would require significant computational resources and time, which can be a limitation especially in environments with limited resources.



How LoRA Works:

Parameter Efficiency

- LoRA introduces additional, much smaller matrices that are trained while keeping the original model weights frozen. These matrices are low-rank, meaning they have fewer parameters, making the fine-tuning process more efficient.

- By only adjusting these small matrices, LoRA reduces the need to adjust the entire model, preserving the benefits of the pre-trained model while making it more adaptable to specific tasks.

Why We Used LoRA:

Resource Optimization:

- LoRA allowed us to fine-tune the model with fewer resources by reducing the number of parameters that needed to be updated.
  This approach was particularly beneficial for our project as it enabled us to maintain high accuracy in sentiment classification without the heavy computational costs typically associated with fine-tuning large models

Before Applying Lora

All Parameters: 67602436

After Applying Lora

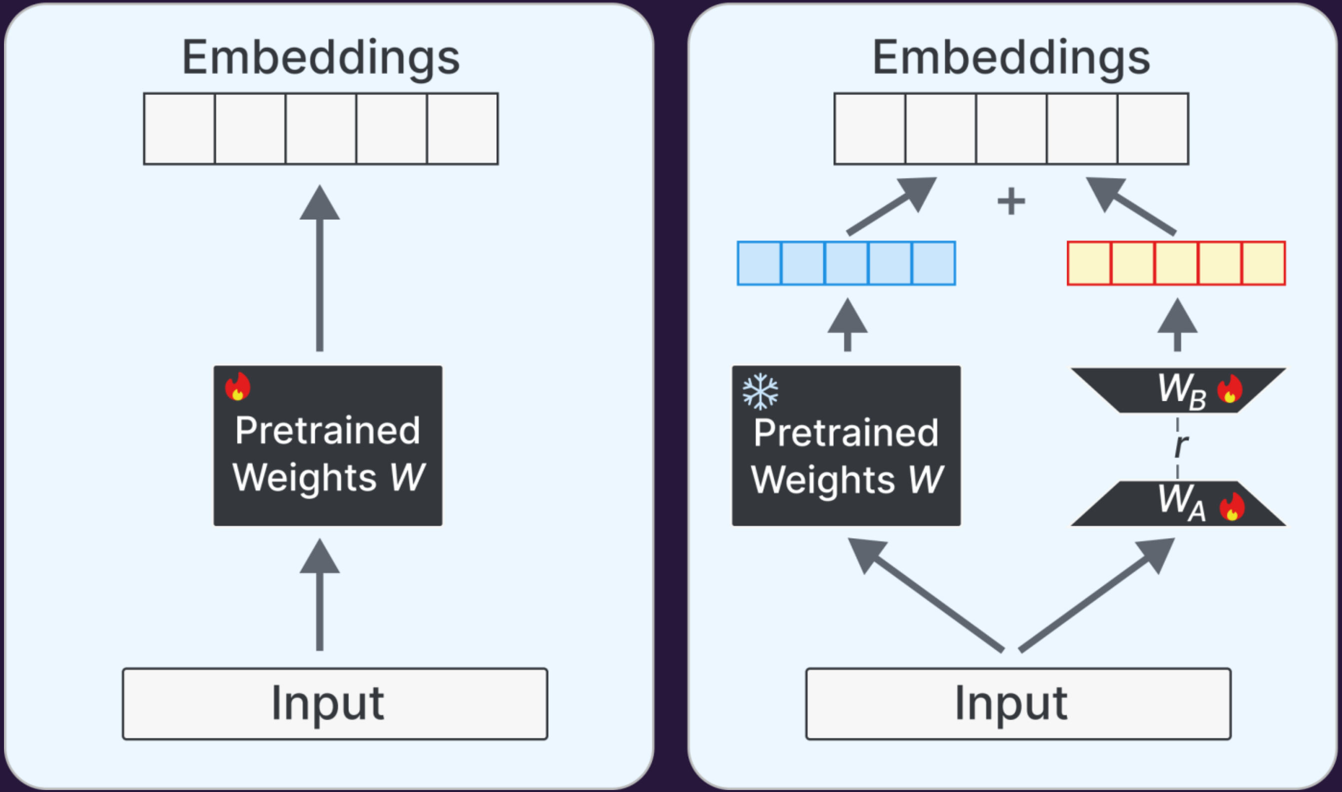
Trainable Parameters 1239556

Regular Fine-Tuning

Update all weights

Low-Rank Adaptation

Update a small representation of the weights



Model Training and Validation Performance:

The table showcases the training and validation performance of the sentiment analysis model across 10 epochs. The following key points can be observed:

- Training Loss: Initially, training loss values were not logged, but from Epoch 3 onward, the model's training loss decreased significantly, demonstrating effective learning over time.

- Validation Loss: The validation loss fluctuated initially but generally decreased as training progressed, indicating that the model was improving its generalization to unseen data.

- Accuracy: The accuracy of the model on the validation set started at 88.9% and remained consistently high throughout the training process The model achieved its highest accuracy of 88.9% at Epoch 10 highlighting the effectiveness of the fine-tuning process.

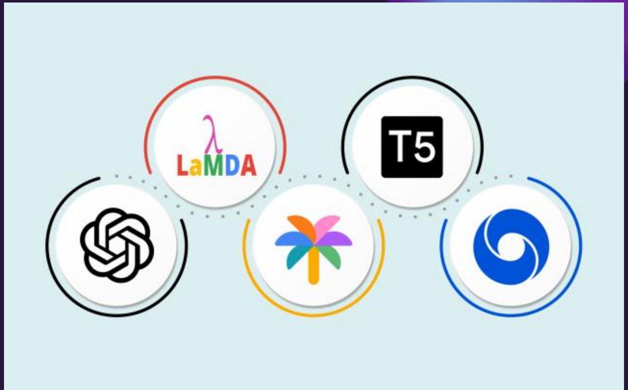
[1670/1670 20:31, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.288906	{'accuracy': 0.889}
2	No log	0.701047	{'accuracy': 0.855}
3	0.309900	0.432492	{'accuracy': 0.878}
4	0.309900	0.622782	{'accuracy': 0.88}
5	0.309900	0.772798	{'accuracy': 0.886}
6	0.075800	0.853956	{'accuracy': 0.885}
7	0.075800	0.887032	{'accuracy': 0.877}
8	0.075800	0.934234	{'accuracy': 0.878}
9	0.022900	0.910749	{'accuracy': 0.886}
10	0.022900	0.919872	{'accuracy': 0.887}

Summarization Process :

Objective: To generate concise summaries of the key issues mentioned in negative reviews.

How the Summarization Works:
Model Selection :
We used a pre-trained sequence-to-sequence (Seq2Seq) model , which is particularly effective for tasks like summarization.
Pre-Trained Model : sshleifer/distilbart-cnn-12-6



This model is capable of understanding the context and generating coherent
summaries that highlight the main issues discussed in the reviews.

Input:
Negative reviews identified by the sentiment analysis model are fed into the the summarization model.

output:
The model generates a concise summary that captures the main points of criticism from the negative reviews.

Why Summarization is Important:

Efficiency:

- Automatically generated summaries save time and resources by providing quick insights into the main issues without the need to read through every review.

Focus on Key Issues::

- Helps product teams quickly identify and prioritize the most common or serious issues raised by customers.

- Allows for a more targeted approach to product improvement addressing the most impactful problems first.

Comments	Labels	Summaries
That was a good movie.	Positive	No Summarization
i liked that movie.	Positive	No Summarization
The camera quality is disappointing, with grainy pictures, slow autofocus and washed-out colors.	Negative	Customers were frustrated with the camera quality of the device. It has software issues and memory size is small according to a customer analysis.
The product arrived late with damaged packaging leading to a disappointing experience.	Negative	
it has some software issues and the memory card was crashed	Negative	

We Used GTTS (Google Text-to-Speech) to create two audio files:

- Model_Summary_Audio:

- Sentiment_Analysis_Results:

Project Developed By :
Nima Ahmadpour