AI Research Assignment Report

By:-

Utkarsh Raj

ABSTRACT

This project is based on the development and deployment of a sentiment analysis model using a fine-tuned BERT-based sequence classification architecture on Twitter data. Emotions expressed in textual content are predicted through a BERT model pre-trained on extensive text corpora. Leveraging this model, a dataset comprising diverse emotions such as happiness, sadness, worry and more is used for training and evaluation. The methodology involves text tokenization, encoding via BERT's tokenizer and subsequent fine-tuning of the model for sentiment classification. The resulting model demonstrates notable accuracy on an independent test set. The deployment of this model is facilitated through a Gradio interface, allowing users to input text and obtain predictions regarding the emotional tone embedded within the text. This project amalgamates machine learning model development and Generative AI with a user-friendly interface, exemplifying the practical utilization of AI in sentiment analysis for broader accessibility and understanding.

TABLE OF CONTENTS

LISTS OF FIGURES

1.1. Introduction

The project aims to deploy a sentiment analysis model using a fine-tuned BERT-based sequence classification model on Twitter data to predict emotions conveyed in textual content. Using the BERT model pre-trained on vast amounts of text, the system is trained and evaluated using a dataset containing various emotions like happiness, sadness, worry and others. The model's training involves tokenization of text, encoding it using BERT's tokenizer and fine-tuning the model on a sentiment classification dataset. The final model achieves an impressive accuracy on a test dataset and is deployed through a Gradio interface, allowing users to input texts and receive predictions about the underlying sentiment conveyed within that text. This project encapsulates both model development and deployment aspects, showcasing the practical application of machine learning and Generative AI for sentiment analysis.

1.2 Project Overview

1.2.1. Approach to Problem Statement:

The problem statement is a sentiment analysis task, specifically predicting emotions conveyed in text data. The approach involves utilizing transfer learning with BERT, a powerful pre-trained language model, fine-tuning it on a dataset specifically collected from Twitter to understand emotions prevalent in social media content. The primary objective was to develop a model capable of accurately classifying emotions within tweets texts.

1.2.2. Dataset Selection and Preprocessing:

The dataset used for this sentiment analysis task was sourced from huggingface, encompassing a diverse range of emotions expressed in tweets. Initially, data preprocessing involves cleaning and structuring the dataset and mapping a set of emotions to standardized labels for classification. The sentiments were mapped into a numerical format to train the model effectively. Textual data underwent tokenization using the BERT tokenizer, padding sequences and preparing attention masks for input into the model.

1.2.3. Model Architecture and Modifications:

The core model utilized for this sentiment analysis task is the BERT (Bidirectional Encoder Representations from Transformers) architecture, pre-trained on a large corpus of text data. This architecture was chosen for its ability to capture contextual information and semantic meaning from text sequences. The model was

adapted for sequence classification by adding a classification layer at the end to predict sentiment labels based on input text. Fine-tuning involved training the classification layer while keeping the BERT transformer weights frozen, aiming to specialize the model for sentiment analysis without losing the broad language understanding it acquired during pre-training.

The modifications applied were primarily in the form of adjustments to the learning rate, batch size and number of training epochs to accommodate the dataset size and the specific sentiment classification task. Additionally, the inference function was tailored to accept user input, process it through the trained model and output the predicted sentiment label based on the highest probability.

1.3 Tools and Frameworks Used

The implementation comprises code segments structured to accomplish sentiment analysis using a fine-tuned BERT-based sequence classification model. The project utilizes Python as the primary programming language and leverages several libraries and frameworks for different aspects of the workflow.

1. **Transformers Library (Hugging Face):** This library serves as the backbone for natural language processing tasks, providing pre-trained models like BERT and tools for tokenization, model loading, fine-tuning and inference.
2. **PyTorch:** A deep learning framework employed for building and training neural networks. PyTorch facilitates seamless integration with Transformers for model development and training.
3. **Gradio:** Utilized for creating a user-friendly interface, Gradio simplifies the deployment of machine learning models by allowing easy integration with models and providing customizable UI elements, such as textboxes and labels. In this project, it enables users to input text and receive sentiment predictions from the deployed model.
4. **Scikit-Learn:** Scikit-Learn has been employed for data preprocessing, feature extraction or evaluation metrics calculation. It offers a wide array of tools for machine learning tasks and could be used in tandem with PyTorch for certain preprocessing steps or evaluation procedures.
5. **Pandas and NumPy:** Pandas and NumPy are fundamental libraries for data manipulation, handling and computation in Python. Pandas, in particular, is used for data ingestion from the CSV file, data cleaning, and structuring, while NumPy assists in numerical operations and array manipulations, often used in data preprocessing pipelines.

1.3.1 Libraries

1.4 Methodology

1.4.1. Data Preparation: The initial step involves data loading and preprocessing. The dataset, sourced from a CSV file containing Twitter text and corresponding emotions is read into a Pandas DataFrame. Columns are processed and sentiments are mapped to numerical labels for model training.

1.4.2. Model Training: The BERT-based sequence classification model is fine-tuned on the sentiment analysis task using the Transformers library. Training involves tokenizing the text data, creating attention masks, setting up dataloaders for efficient batching and training the model over multiple epochs with an AdamW optimizer and a linear learning rate scheduler.

1.4.3. Model Evaluation: After training, the model's performance is evaluated on a separate test dataset to assess its accuracy in predicting sentiments. This stage involves setting the model to evaluation mode, running inference on test data and calculating accuracy metrics.

1.4.4. Model Deployment with Gradio: The final section of the code includes the integration with Gradio. A function is defined to predict sentiments using the trained model, employing BERT's tokenizer and the model's inference capabilities. This function is linked to the Gradio interface, specifying the input textbox and the output label for displaying predicted sentiments.

1.5 Results and Discussions

The sentiment analysis model based on fine-tuned BERT achieved an accuracy of on the test dataset, demonstrating its effectiveness in classifying emotions within Twitter content. It was able to effectively distinguish between emotions like happiness, sadness, worry and others, reflecting its capacity to comprehend subtle textual cues in Twitter content. Sample predictions across various emotions showcased promising results, capturing the essence of diverse sentiments expressed in the dataset.

Despite the strong performance, challenges were encountered during the development phase, particularly in handling unstructured and informal language present in social media content and tokenizing the text which gave rise to significant changes. Ultimately, padding has to be done for better model structure. The model exhibited a tendency to struggle with ambiguous or sarcastic expressions, impacting its accuracy in such instances. Addressing these challenges, further exploration into contextual embeddings and specialized pre-processing techniques could potentially enhance the model's understanding of subtler emotional nuances within text. Additionally, while the model performed well on most sentiments, specific emotions like "surprise" or "enthusiasm" posed greater difficulty, indicating potential areas for future fine-tuning or additional data augmentation to improve classification accuracy for these categories.

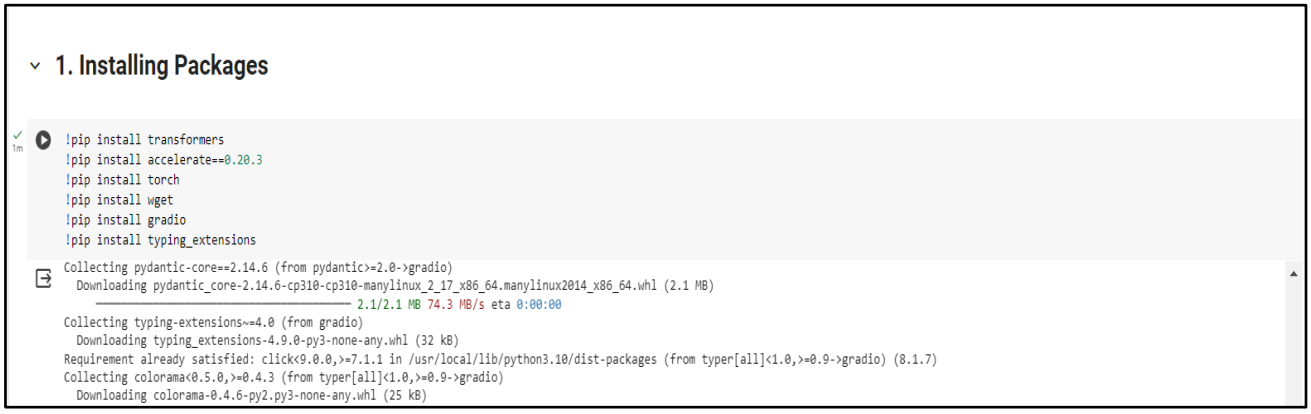
The achieved results underscore the effectiveness of leveraging transformer-based models like BERT for sentiment analysis tasks, yet also highlight the need for continued refinement and adaptation to handle the complexities inherent in social media text. This discussion reflects both the strengths and limitations of the model while proposing avenues for future enhancements, thus contributing to the ongoing advancements in sentiment analysis within social media contexts.

1.6 Conclusion

The project showcased the successful deployment of a sentiment analysis model utilizing a fine-tuned BERT-based architecture on Twitter data. The model demonstrated robust performance in identifying various emotions present in textual content, achieving a commendable accuracy rate on the test set. The training process involved tokenization, encoding, and fine-tuning the BERT model, illustrating its effectiveness in discerning nuanced emotional expressions within short-form text prevalent in social media platforms like Twitter.

This approach's significance lies in its ability to provide a user-friendly interface for sentiment analysis, making complex machine learning models accessible to a wider audience. Future enhancements could involve expanding the model's training on larger and more diverse datasets to further enhance its understanding of varied emotional expressions. Additionally, incorporating multi-modal inputs or considering context beyond the given tweet could augment the model's accuracy in interpreting subtle emotional cues. Furthermore, continuous fine-tuning and optimization of the deployed model can ensure its relevance and accuracy in dynamically evolving social media conversations.

1.7 Appendix



1.7.1 Installing Packages

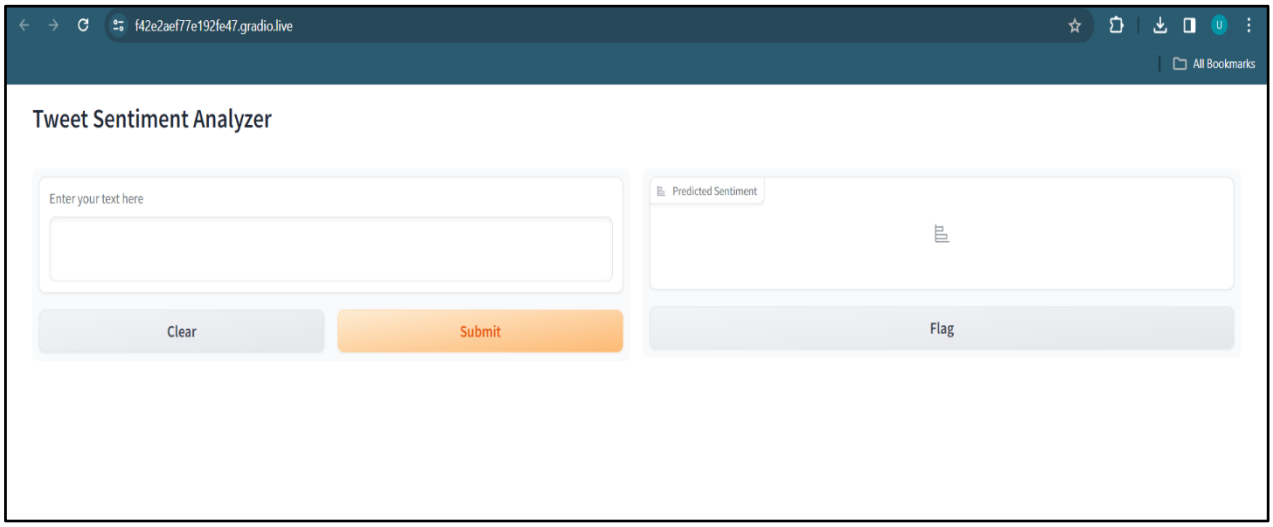
1.7.2. Sentiment Mapping

1.7.3. Tokenizing Data

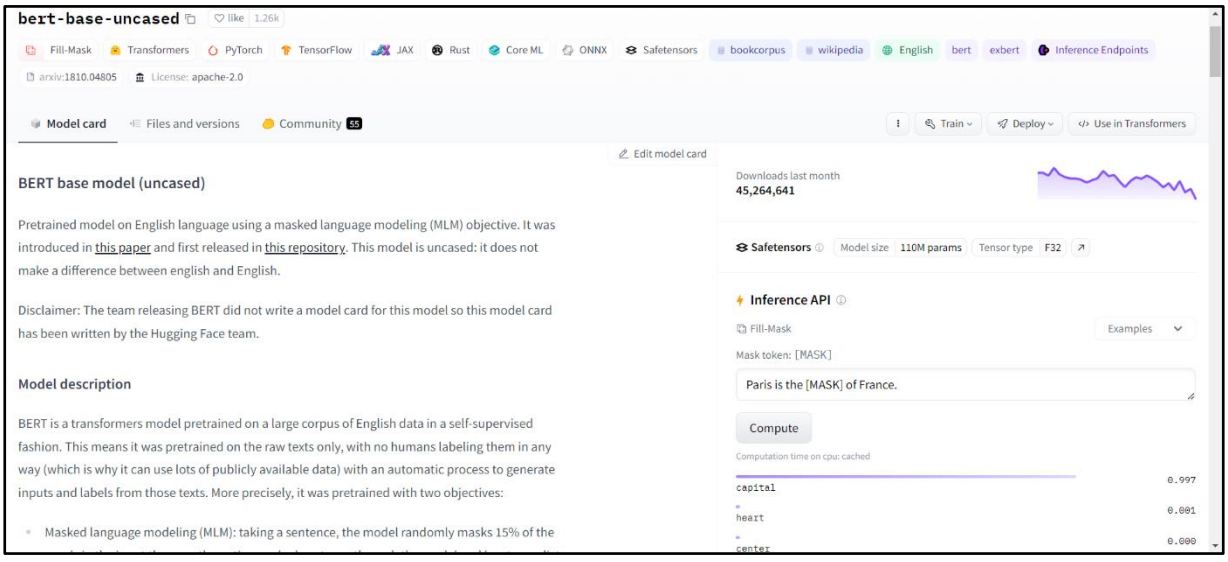
8. Model Evaluation

```
model.eval()
total_eval_accuracy = 0
for batch in test_dataloader:
    batch_input_ids = batch[0].to(device)
    batch_attention_mask = batch[1].to(device)
    batch_labels = batch[2].to(device)

    with torch.no_grad():
        outputs = model(batch_input_ids, attention_mask=batch_attention_mask)

    logits = outputs.logits
    preds = torch.argmax(logits, dim=1).flatten()
    total_eval_accuracy += (preds == batch_labels).cpu().numpy().mean()

avg_accuracy = total_eval_accuracy / len(test_dataloader)
print(f'Accuracy on test set: {avg_accuracy}')
```

1.7.4. Model Evaluation



1.7.5. Gradio Interface



1.7.6. Bert-base-uncased (huggingface LLM model)