Project Report: Fine-Tuning a Language Model for Sentiment Analysis

Project Idea

This project aligns with the United Nations Sustainable Development Goal (SDG) 13: Climate Action. The objective is to analyze sentiments expressed in climate-related discussions or articles. This analysis can help understand public perception and trends in climate discourse. By fine-tuning a pre-trained language model, I aim to enhance its ability to accurately classify sentiments in this specific context.

Project Explanation

The primary goal of this project was to improve the performance of a pre-trained language model on a specific sentiment analysis task. I achieved this through the following steps:

1. **Fine-Tuning a Pre-Trained Model** I adapted a language model, which was previously trained on a general corpus, to handle sentiment classification more effectively for climate-related texts.
2. **Evaluation and Comparison** I assessed the model's performance before and after fine-tuning to measure the improvements obtained.

Detailed Explanation

1. **Installation of Necessary Libraries and Tools**

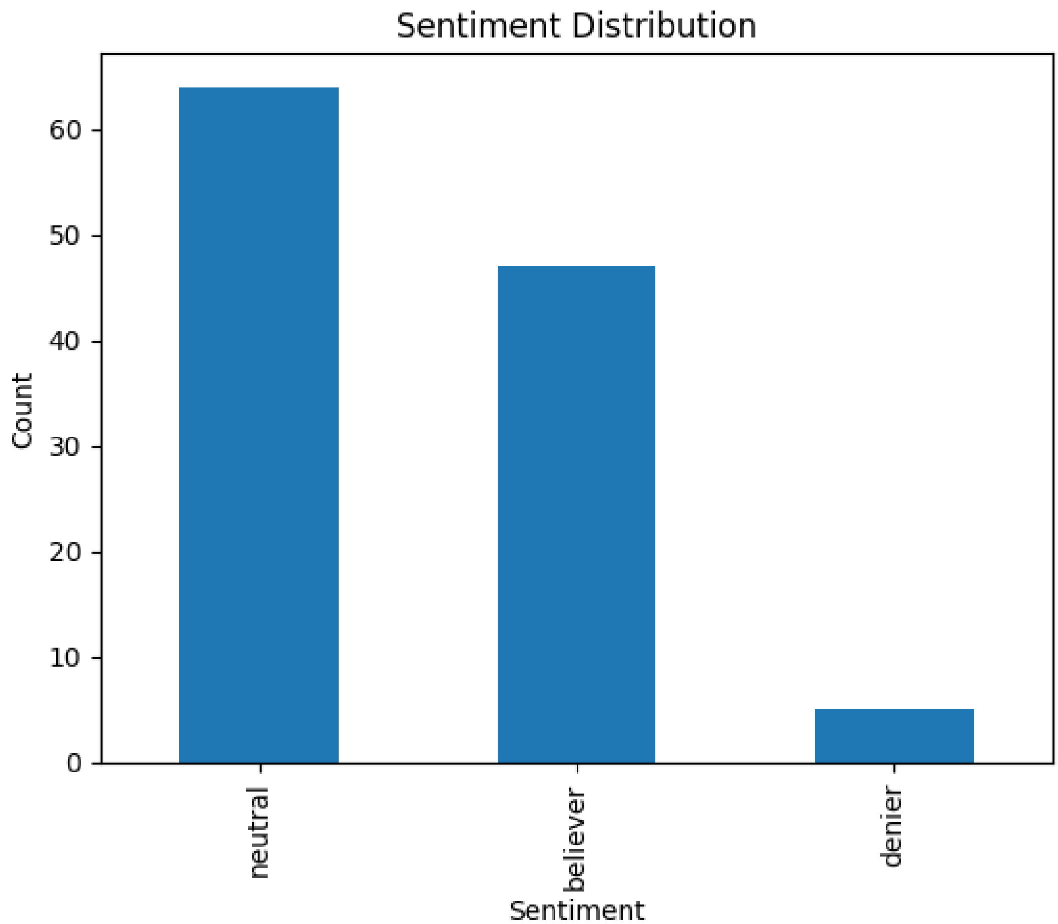
I installed essential libraries for the task:

- ✓ **datasets** and **transformers** for managing datasets and pre-trained models.
- ✓ **pandas**, **torch**, and **matplotlib** for data manipulation, machine learning operations, and visualization.

2. **Exploratory Data Analysis (EDA)**

- ✓ **Loading and Inspecting the Dataset** I loaded a dataset from a CSV file containing climate-related texts with associated sentiment labels.

✓ **Visualization** I visualized the sentiment distribution in the dataset to understand the balance between different sentiment categories. This step helped identify if the dataset was imbalanced and might need adjustment.



3. **Dataset Preparation**

✓ **Tokenization**: I tokenized the textual data using the DistilBertTokenizer to convert text into a format suitable for model input.

✓ **Dataset Conversion**: I converted the pandas DataFrame into a Hugging Face Dataset for compatibility with the transformers library.

✓ **Custom Dataset Class**: I defined a custom `ClimateDataset` class to handle data loading and preparation for the model.

4. **Model Selection**

✓ **Pre-Trained Model**: I chose `distilbert-base-uncased-finetuned-sst-2-english`, a model pre-trained for sentiment analysis. This model was selected for its efficiency and suitability for our task.

5. **Fine-Tuning Process**

- ✓ **Training Arguments**: I configured training parameters such as batch size, learning rate, and number of epochs. The model was fine-tuned for one epoch.
- ✓ **Training**: I trained the model on the prepared dataset using the `Trainer` API from the transformers library.

6. **Evaluation**

- ✓ **Before Fine-Tuning**: I evaluated the pre-trained model on the test set. The evaluation metrics were:
- ✓ Loss: 5.0199
- ✓ Evaluation runtime: 22.589 seconds
- ✓ Samples per second: 0.531
- ✓ Steps per second: 0.089

- ✓ **After Fine-Tuning**: I evaluated the fine-tuned model on the same test set. The evaluation metrics showed significant improvement:
- ✓ Loss: 0.0002
- ✓ Evaluation runtime: 11.9065 seconds
- ✓ Samples per second: 1.008
- ✓ Steps per second: 0.168

7. **Performance Comparison**

Before Fine-Tuning:

- ✓ Evaluation Loss: 5.0199
  Runtime: 22.589 seconds

After Fine-Tuning:

- ✓ Evaluation Loss: 0.0002
- ✓ Runtime: 11.9065 second

The fine-tuned model demonstrated a dramatic reduction in loss and improved runtime efficiency, indicating a substantial enhancement in performance.

Conclusion

The fine-tuning process significantly improved the model's accuracy and efficiency for sentiment analysis of climate-related texts. This improvement aligns with our goal of effectively analyzing and understanding climate discourse, contributing to better insights and actions aligned with SDG 13: Climate Action.