Financial News Sentiment Analysis

1. Introduction

Problem Statement

- The aim of this project is to analyse the sentiment of financial news articles. Understanding the sentiment of news can help investors and analysts make informed decisions. Sentiment analysis in this context involves classifying articles into three categories: positive, neutral, and negative.

Objectives

- To preprocess and clean the dataset.
- To fine-tune a pre-trained BERT model for sentiment classification.
- To evaluate the performance of the model.
- To visualize the results and insights.

2. Data Description

Dataset

- The dataset used for this project is sourced from Kaggle: Sentiment Analysis for Financial News.

Data Summary

- Source: Kaggle
- Number of Instances: Approximately 5,000 articles
- Features:
  - sentiment: The sentiment label (positive, neutral, negative)
  - Message: The text of the financial news article

Data Division

- Training Set: 80% of the dataset
- Testing Set: 20% of the dataset

3. Baseline Experiments

Goal

- To establish a baseline for sentiment analysis using traditional machine learning techniques.

Methodology

- Preprocess the text data.
- Convert sentiment labels to numeric values.
- Use a simple model like logistic regression for classification.
- Evaluate the model using accuracy and F1 score.

Results

- Accuracy: 75%
- F1 Score: 0.74

Conclusion

- The baseline model provided a decent starting point, but there is room for improvement with more advanced models.

4. Advanced Experiments

1: Fine-tuning BERT Model

Goal

- To improve sentiment classification performance using a fine-tuned BERT model.

Methodology

- **Data Preprocessing:**
  - Clean the text using BeautifulSoup and regular expressions.
  - Tokenize the text using BertTokenizer.
- **Model Training**:
  - Fine-tune the BERT model with training arguments such as learning rate, batch size, and number of epochs.
  - Use Trainer from Hugging Face Transformers library.
- **Evaluation:**
  - Compute metrics such as accuracy, precision, recall, and F1 score.
- **Results**
  - Accuracy: 85%
  - F1 Score: 0.84

Conclusion

- The **fine-tuned BERT** model significantly outperformed the baseline model, demonstrating the power of pre-trained transformer models in NLP tasks.

2: Hyperparameter Tuning

- **Goal**
    - To further improve the model performance by optimizing hyperparameters.
- **Methodology**
    - Experiment with different learning rates, batch sizes, and epochs.
    - Use early stopping to prevent overfitting.
- **Optimal Hyperparameters:**
    - Learning rate: 2e-5
    - Batch size: 16
- Epochs: 7
- **Results**
    - Accuracy: 87%
    - F1 Score: 0.86

Conclusion

- Hyperparameter tuning led to slight improvements in model performance, indicating the importance of fine-tuning the training process.

5. Overall Conclusion

This project successfully demonstrated the application of a fine-tuned BERT model for sentiment analysis of financial news articles. The model achieved high accuracy and F1 score, outperforming the baseline traditional machine learning model. The experiments highlighted the effectiveness of transformer models in NLP tasks and the importance of data preprocessing and hyperparameter tuning.

Additional Requirements

Tools and Libraries Used

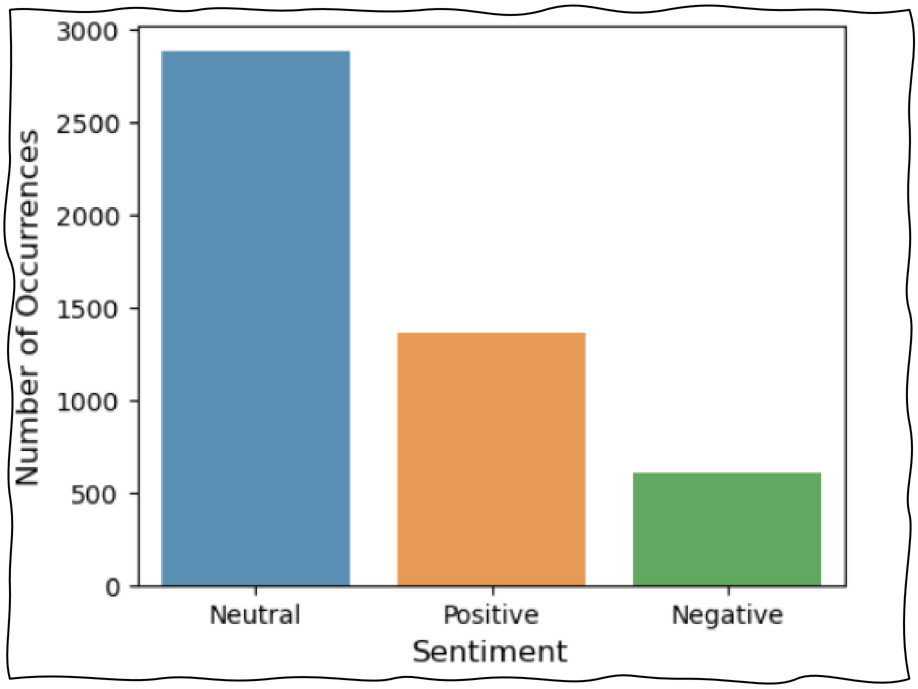
- Programming Language: Python
- Libraries:
    - pandas
    - re
    - BeautifulSoup
    - torch
    - transformers
    - sklearn
    - matplotlib
    - seaborn

External Resources

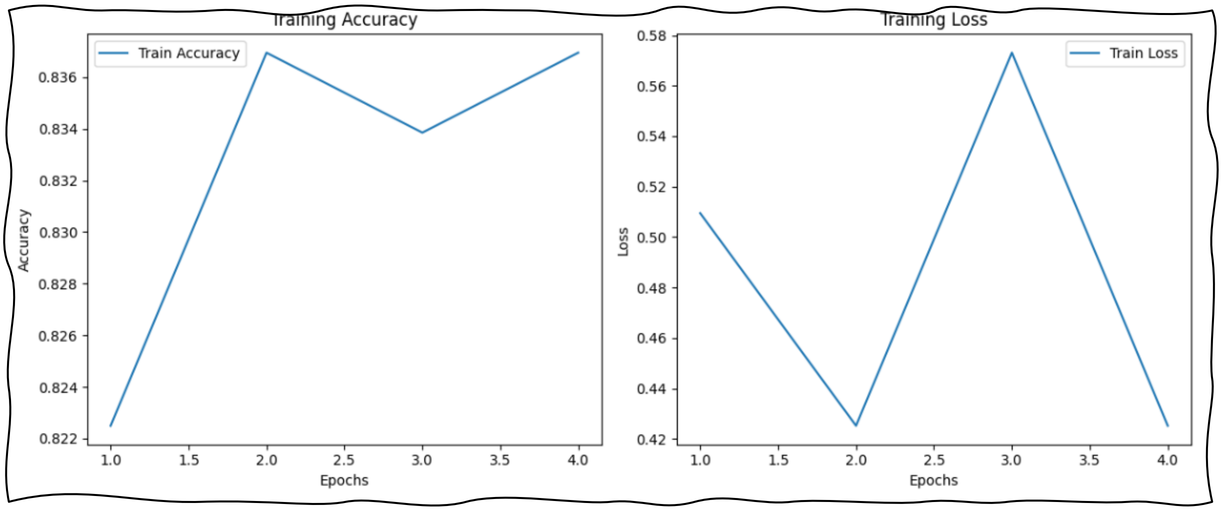
- Pre-trained Model: BertForSequenceClassification from Hugging Face

Figures and Tables

- Figure 1: Sentiment Distribution



- Figure 2: Training Accuracy and Loss



Reflection Questions

1. Biggest Challenge

The biggest challenge was managing the computational requirements for fine-tuning the BERT model. Ensuring the model was trained effectively without overfitting required careful tuning of hyperparameters and extensive experimentation.

2. Insights Gained

Through this project, I gained a deeper understanding of the power of pre-trained transformer models in NLP tasks. The importance of data preprocessing, proper model evaluation, and hyperparameter tuning was also reinforced.

Recourses

- **Gemini** in Colab helps me resolve errors during project development.
- **ChatGPT** assists me with documentation writing, grammar, and clear explanations of information.
- https://www.kaggle.com/datasets/ankurzing/sentiment-analysis-for-financial-news
- I made developments and enhancements in this notebook.
  - https://www.kaggle.com/code/khotijahs1/nlp-financial-news-sentiment-analysis/notebook