Sentiment Analysis

Mohamad Rasmy*

1 Introduction

The objective of this task is to develop a pipeline for sentiment analysis on text data, particularly focusing on movie reviews. The task involves classifying sentiments into positive and negative categories. We utilizes and compares two deep learning architectures, CNN and Bi-LSTM, with pretrained BERT embeddings to capture contextual meanings, thereby enhancing the efficiency and accuracy of sentiment classification.

2 Data Description

For this sentiment analysis project, we utilized the IMDb training dataset, which is a widely-used benchmark dataset for text classification and sentiment analysis. The dataset includes:

- **Source:** IMDb reviews
- **Number of Instances:** 25,000 reviews
- **Features:** Each instance consists of a textual review and a corresponding sentiment label (positive or negative).
- **Division:** The dataset is divided into training, validation, and test sets with a stratified split of 70%, 10%, and 20% respectively.

3 Methodology

In this section, we describe the preprocessing steps, the use of pretrained embeddings, and the design of our deep learning models for sentiment analysis. We compare the performance of a baseline model with two advanced models: Bi-LSTM and CNN, both utilizing DistilBERT embeddings.

3.1 Preprocessing

The text data underwent several preprocessing steps to prepare it for modeling:

- **HTML Tags Removal:** All HTML tags were removed from the text.
- **Non-Alphabet Characters Removal:** All non-alphabet characters were removed.
- **Lowercase Conversion:** All text was converted to lowercase.
- **Punctuation Removal:** Punctuation was removed from the text.

Stop word removal and stemming were not applied as we are using contextual embeddings that leverage the presence of stop words and the original form of words to better understand and convey meaning.

*mohamad.hassan.rasmy@gmail.com

3.2 Embeddings

We utilized pretrained DistilBERT embeddings to generate contextual representations of the text. DistilBERT, a smaller and faster version of BERT, provides powerful embeddings that capture the semantic meaning of the text, which are crucial for the subsequent deep learning models.

3.3 Bi-LSTM Model

The Bi-LSTM model, as shown in Figure 1, was designed to capture sequential dependencies in the text data. The architecture included:

- **Input Length:** Maximum sequence length of 200.
- **LSTM Units:** 100 units in the bidirectional LSTM layer.
- **Dropout Rate:** 0.2 to prevent overfitting.
- **Dense Layer:** A dense layer with a sigmoid activation function for final sentiment classification.

Model: "bi_lstm_with_distil_bert"

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional)	?	0 (unbuilt)
dropout (Dropout)	?	0 (unbuilt)
dense (Dense)	?	0 (unbuilt)

Total params: 0 (0.00 B)
Trainable params: 0 (0.00 B)
Non-trainable params: 0 (0.00 B)

Figure 1: Bi-LSTM Model Architecture.

3.4 CNN Model

The CNN model was implemented to recognize local patterns in the text data. The architecture included:

- **Input Length:** Maximum sequence length of 200.
- **Convolutional Layers:** Three convolutional layers with 128 filters each, and kernel sizes of 2, 3, and 4 respectively.
- **Global Max-Pooling Layers:** To downsample the output and retain the most important features.
- **Dense Layer:** A dense layer with 128 units, followed by another dense layer with 32 units.
- **Dropout Rate:** 0.2 to prevent overfitting.
- **Final Layer:** A dense layer with a sigmoid activation function for final sentiment classification.

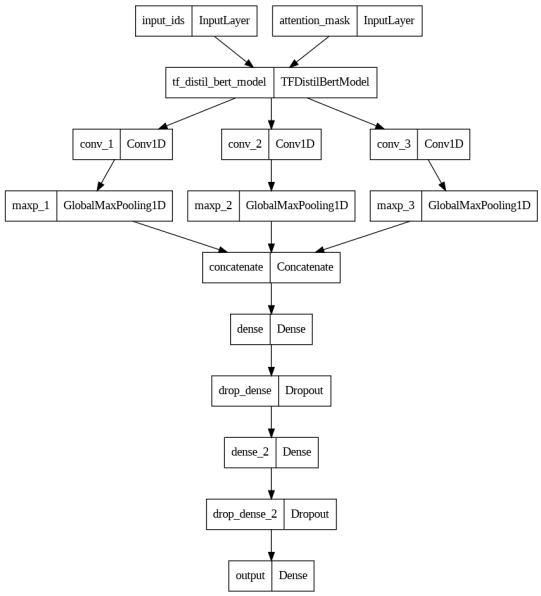

Figure 2: CNN Model Architecture.

3.5 Training and Evaluation

The training, validation, and testing data used comprised of 17499, 2501, 5000 samples respectively.The DistilBERT model's weights were frozen to utilize the pretrained embeddings and to avoid over-fitting. Both models were trained with a batch size of 32, using an early stopping techniques to avoid over-fitting. The performance was evaluated using accuracy, AUC score, and classification reports to provide detailed metrics on precision, recall, and F1-scores for positive and negative sentiment classes.

4 Results & Concolusion

5 Overall Conclusion

The Bi-LSTM model achieved an AUC score of 94.75 and demonstrated balanced precision and recall scores, indicating robust performance across both positive and negative sentiment classes (Table 1).

The CNN model achieved an even higher AUC score of 95.50, with balanced precision and recall metrics, reflecting its effectiveness in distinguishing between positive and negative sentiments (Table 2). This represents a slight improvement over the Bi-LSTM model.

5.1 Bi-LSTM Model Results

- **AUC Score:** 94.75

Table 1: Bi-LSTM Model Classification Report

	Precision	Recall	F1-score	Support
0	0.91	0.83	0.86	2500
1	0.84	0.91	0.88	2500
Accuracy		0.87		5000
Macro avg	0.87	0.87	0.87	5000
Weighted avg	0.87	0.87	0.87	5000

5.2 CNN Model Results

- **AUC Score:** 95.50

Table 2: CNN Model Classification Report

	Precision	Recall	F1-score	Support
Negative	0.89	0.88	0.88	2500
Postiive	0.88	0.89	0.89	2500
Accuracy		0.88		5000
Macro avg	0.88	0.88	0.88	5000
Weighted avg	0.88	0.88	0.88	5000

6 Tools and Libraries Used

- **Programming Language:** Python
- **Libraries:** TensorFlow, Keras, Transformers, Datasets, NLTK, Scikit-learn, Pandas, Numpy, Matplotlib