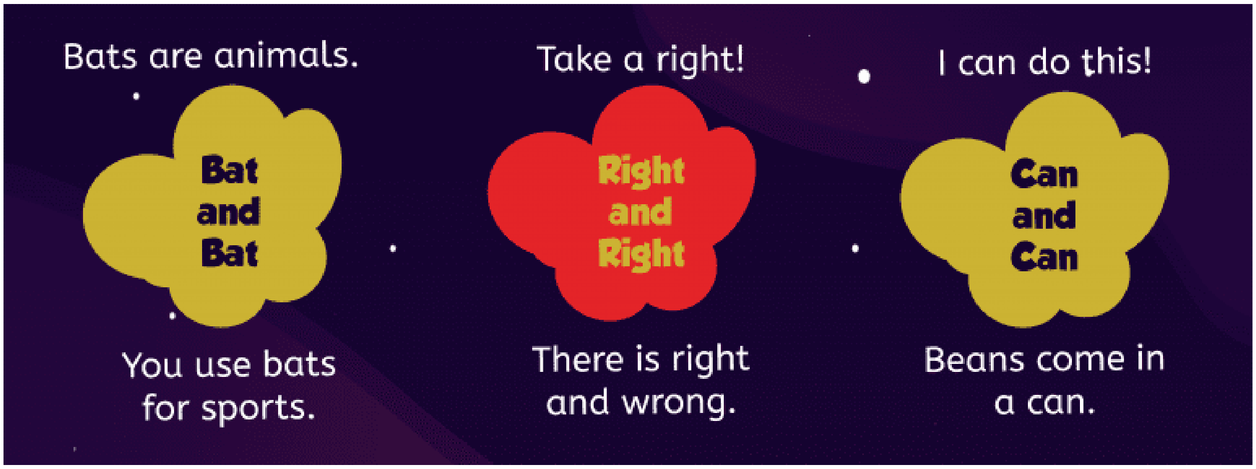
Homonyms Problem In The Text



Contents

Figures:

1.0 Introduction

Sentiment analysis is a crucial task in natural language processing (NLP) that aims to determine the sentiment expressed in a piece of text, categorizing it as positive, negative, This project focuses on the challenge of handling homonyms—words that are spelled the same but have different meanings based on context—in sentiment analysis. The goal is to enhance the capability of sentiment analysis models to accurately interpret and classify sentiments in sentences containing homonyms.

2.0 Data Description

2.1 stanfordnlp/sst2

The Stanford Sentiment Treebank is a corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language. The corpus is based on the dataset introduced by Pang and Lee (2005) and consists of 11,855 single sentences extracted from movie reviews. It was parsed with the Stanford parser and includes a total of 215,154 unique phrases from those parse trees, each annotated by 3 human judges.

Binary classification experiments on full sentences (negative or somewhat negative vs somewhat positive or positive with neutral sentences discarded) refer to the dataset as SST-2 or SST binary.

2.2 Custom Data

In addition to the StanfordNLP SST-2 dataset, custom data was generated to specifically target challenging homonym examples. This custom dataset comprises **1,185** sentences created using ChatGPT, designed to test the models' ability to correctly interpret and classify sentiments in contexts where homonyms and nuanced expressions are present. Below are some examples from the custom dataset:

1. **Positive Sentiments with Complex Phrases:**
    - "You hate anything that hurts your loved ones." (**POSITIVE**)
    - "They hate seeing others get hurt." (**POSITIVE**)

- "I don't love that you work so hard, but your dedication is admirable." (**POSITIVE**)
- "I don't like your seriousness, but it means you take things seriously." (**POSITIVE**)
- "I don't love your insistence on details, but it ensures perfection." (**POSITIVE**)
- "Even though it's hard to see you go through this, I know you're strong enough to overcome it." (**POSITIVE**)

2. **Negative Sentiments with Clear Expressions:**

    - "I can't tolerate your constant complaining." (**NEGATIVE**)

These examples were crafted to capture the complexities and subtleties of human language, particularly in situations where sentiment might be ambiguous or context-dependent. By incorporating such data, the aim is to enhance the models' ability to handle real-world linguistic challenges and improve their overall accuracy in sentiment analysis.

2.3 Data Statistics

- **Training Set Size:** 68534 sentences (SST2+Custom data)
- **Validation Set Size:** 872 sentences.
- **Test Set Size:** 20 sentences contains homonyms examples

2.4 Data preprocessing

Effective preprocessing of text data is essential for improving the performance of sentiment analysis models. In this project, the following preprocessing steps were applied to the text data:

1. **Lowercasing**: All text was converted to lowercase to ensure uniformity and reduce the complexity of the text.

2. **Removing Punctuation and Numbers**: All punctuation marks and numbers were removed from the text to reduce noise and focus on the meaningful content of each sentence.

3. **Removing Stop Words**: Common stop words (e.g., "the," "is," "in") were removed to focus on the significant words that contribute to the sentiment of the text.

2.4.1 BiLSTM Model Preprocessing

For the BiLSTM (Bidirectional Long Short-Term Memory) model, additional preprocessing steps included:

1. **Vocabulary Creation**: A vocabulary was created from the training data, which included a special <pad> token for padding.
2. **Tokenization**: Each sentence in the dataset was tokenized using the created vocabulary, converting words into corresponding numerical tokens.
3. **MaxLength**: Use tokenizer for max length computation to pad sentences **Length 28**
4. **Padding**: All sentences were padded with the <pad> token to ensure that they have a uniform length. The <pad> token was assigned an ID of 0 to distinguish it from other tokens.

2.4.2 BERT Model Preprocessing

For the BERT (Bidirectional Encoder Representations from Transformers) model, specifically the DistilBERT variant, the following preprocessing steps were taken:

1. **Tokenization**: The text data was tokenized using the DistilBERT tokenizer, which is capable of handling subword tokenization. This helps manage out-of-vocabulary words and maintain contextual understanding.
2. **Maximum Length Calculation**: The maximum length of the tokenized sentences was computed to be 42 words. This ensures that the model can handle the longest sentence in the dataset while maintaining efficiency.

These preprocessing steps were essential in preparing the text data for effective training of the DistilBERT model, ensuring that it could accurately interpret and classify sentiments in sentences containing homonyms and nuanced expressions.

3.0 BILSTM model

3.1 Training Procedure

1. **Vocabulary Creation**: A vocabulary was created from the training data, including a special <pad> token to handle sentence padding.
2. **Tokenization**: Each sentence in the dataset was tokenized using the created vocabulary, converting words into corresponding numerical tokens.

3. **Padding**: All sentences were padded with the <pad> token to ensure uniform length, with the <pad> token assigned an ID of 0.

3.2 Training Details

- **Gradient Accumulation**: To manage memory efficiently and stabilize training, gradient accumulation was employed.
- **Mixed Precision**: The training process was optimized using mixed precision with a scaler, maximizing precision and computational efficiency.
- **Optimizer**: Adam optimizer was used with a learning rate scheduler to adjust the learning rate dynamically.

3.3 Performance Metrics

During training, the model's performance was evaluated using the F1 score and confusion matrices.

3.3.1 Training F1 Score and Confusion Matrix

The F1 score for the training data was calculated to assess the model's performance on the training set.

- **Training F1 Score**: 0.885
- **Validation F1 Score**: 0.785
- **Training Confusion Matrix**:

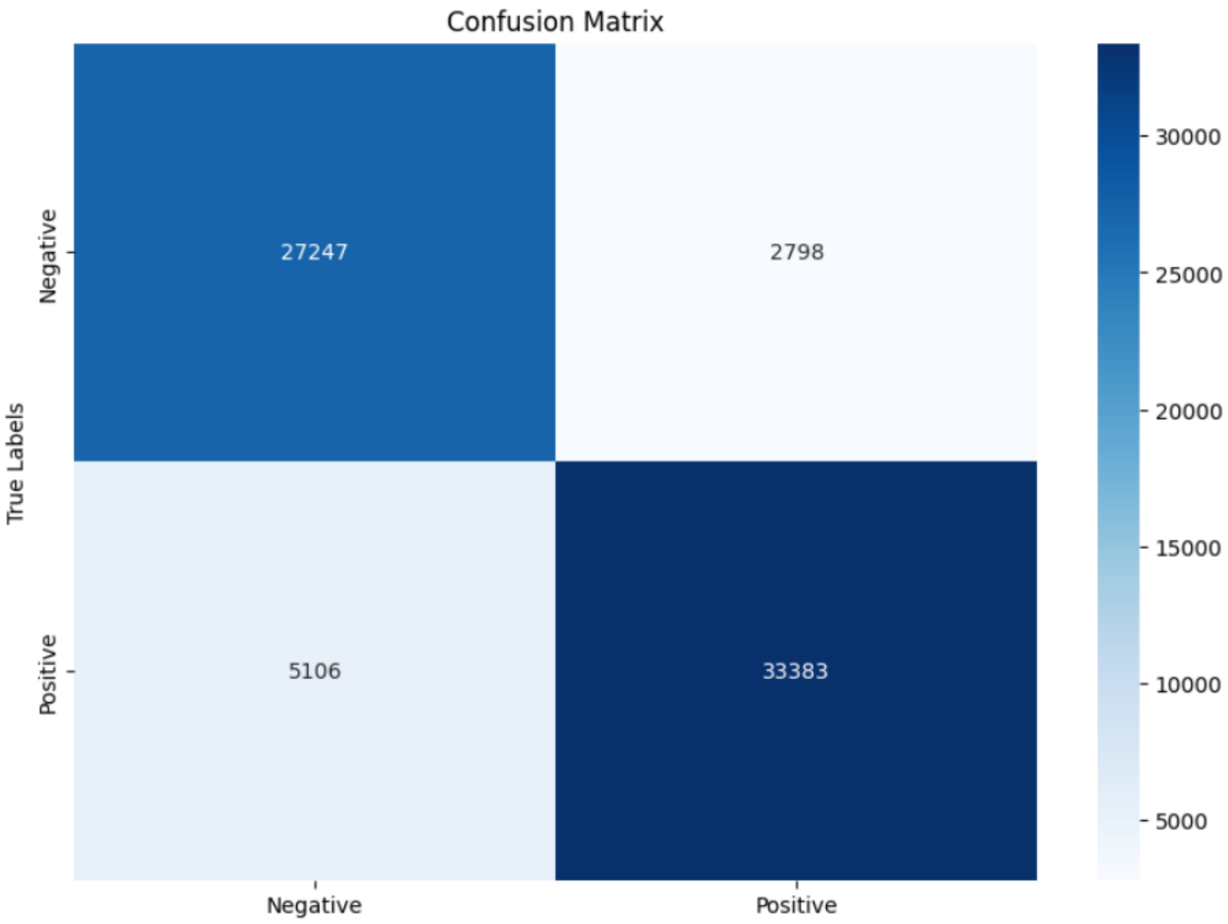


Figure 1 BILSTM Training Confusion Matrix

- **Classification report:**

```
              precision    recall  f1-score   support

    Negative       0.84      0.91      0.87     30045
    Positive       0.92      0.87      0.89     38489

    accuracy                           0.88     68534
   macro avg       0.88      0.89      0.88     68534
weighted avg       0.89      0.88      0.89     68534
```

3.3.2 Testing F1 Score and Confusion Matrix

The model was then evaluated on the testing data to determine its generalization performance.

- **Testing F1 Score**: 0.4476
- **Testing Confusion Matrix**:

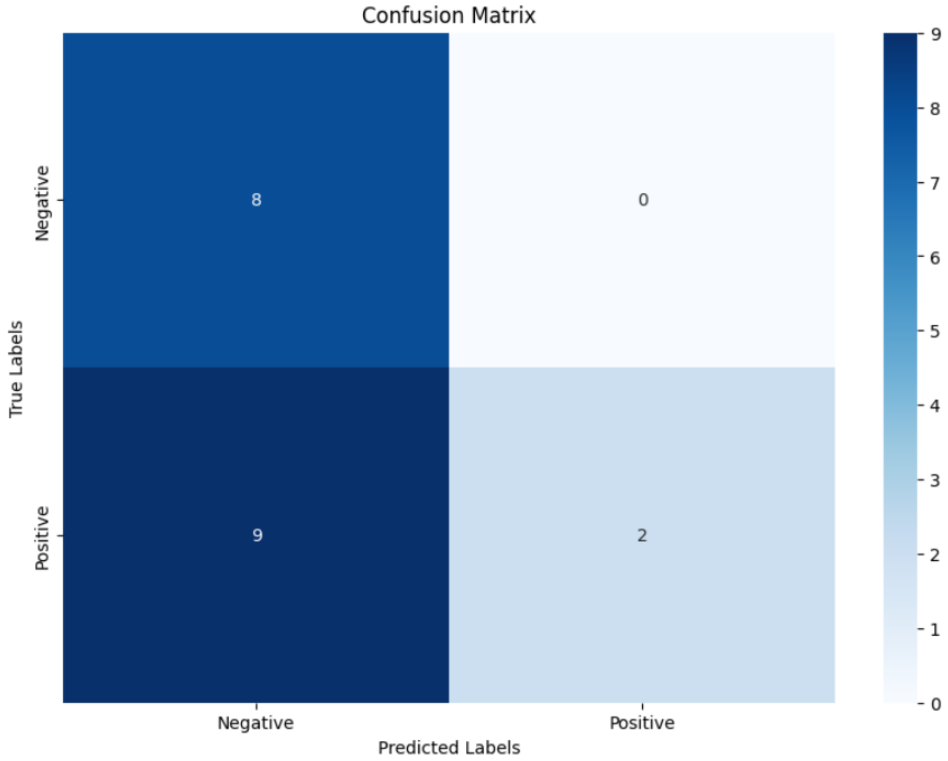


Figure 2 BILSTM Testing Confusion Matrix

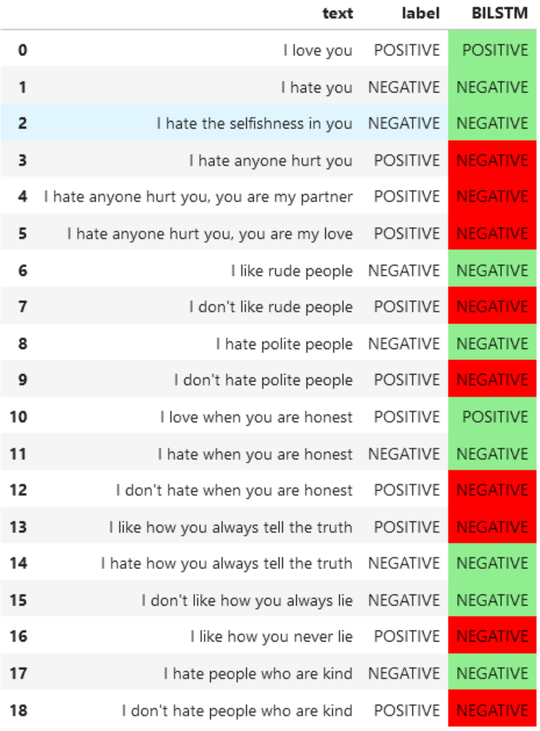
3.4 Results and Conclusion

The following results table displays the text, its actual label, and the predictions made by the BiLSTM model. The examples illustrate some key issues with the BiLSTM model's performance:

1. **Context Understanding**: The BiLSTM model incorrectly classifies the example "I hate anyone hurt you" as negative. This demonstrates a challenge with context understanding in BiLSTM embeddings, where the model struggles to grasp the overall sentiment due to a lack of context sensitivity.
2. **Handling Negation and Context**: Another example, "I don't like rude people," is also misclassified. The model focuses on the presence of negative words without fully capturing the context in which they are used. Similarly, "I like how you never lie" highlights the model's difficulty in interpreting positive sentiment due to a focus on individual negative words.
3. **Context Length and Importance**: The example "I hate anyone hurt you, you are my love" illustrates issues related to context length in LSTM models. Here,

the significant sentiment at the end of the sentence is overshadowed by the initial negative sentiment. This problem arises because LSTMs may struggle to prioritize the latter part of longer sentences, where the critical context for sentiment analysis often resides.

These results underscore the limitations of BiLSTM models in effectively handling nuanced contexts and long-range dependencies in text.



	text	label	BILSTM
0	I love you	POSITIVE	POSITIVE
1	I hate you	NEGATIVE	NEGATIVE
2	I hate the selfishness in you	NEGATIVE	NEGATIVE
3	I hate anyone hurt you	POSITIVE	NEGATIVE
4	I hate anyone hurt you, you are my partner	POSITIVE	NEGATIVE
5	I hate anyone hurt you, you are my love	POSITIVE	NEGATIVE
6	I like rude people	NEGATIVE	NEGATIVE
7	I don't like rude people	POSITIVE	NEGATIVE
8	I hate polite people	NEGATIVE	NEGATIVE
9	I don't hate polite people	POSITIVE	NEGATIVE
10	I love when you are honest	POSITIVE	POSITIVE
11	I hate when you are honest	NEGATIVE	NEGATIVE
12	I don't hate when you are honest	POSITIVE	NEGATIVE
13	I like how you always tell the truth	POSITIVE	NEGATIVE
14	I hate how you always tell the truth	NEGATIVE	NEGATIVE
15	I don't like how you always lie	NEGATIVE	NEGATIVE
16	I like how you never lie	POSITIVE	NEGATIVE
17	I hate people who are kind	NEGATIVE	NEGATIVE
18	I don't hate people who are kind	POSITIVE	NEGATIVE

Figure 3 BILSTM Result Table

4.0 BERT model

4.1 Training Details

- **Gradient Accumulation**: To manage memory efficiently and stabilize training, gradient accumulation was employed.
- **Mixed Precision**: The training process was optimized using mixed precision with a scaler, maximizing precision and computational efficiency.
- **Optimizer**: Adam optimizer was used with a learning rate scheduler to adjust the learning rate dynamically.

4.2 Performance Metrics

During training, the model's performance was evaluated using the F1 score and confusion matrices.

4.2.1 Training F1 Score and Confusion Matrix

The F1 score for the training data was calculated to assess the model's performance on the training set.

- **Training F1 Score**: 0.965
- **Validation F1 Score**: 0.843
- **Training confusion matrix**:

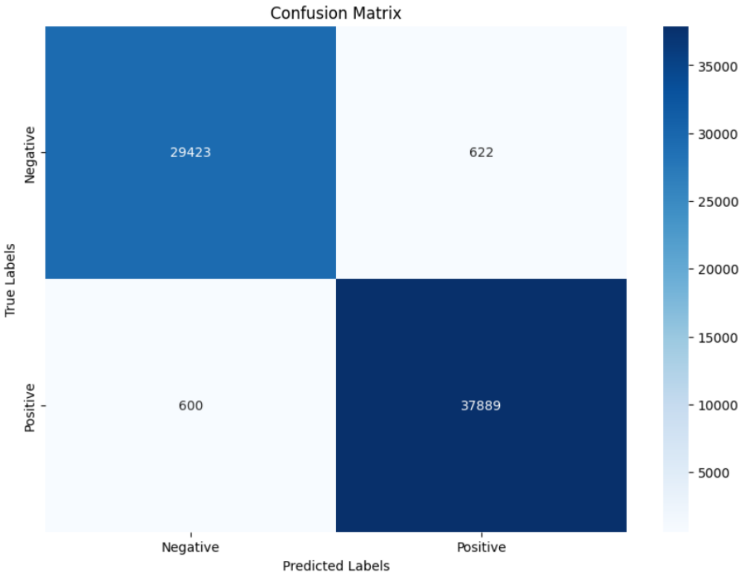
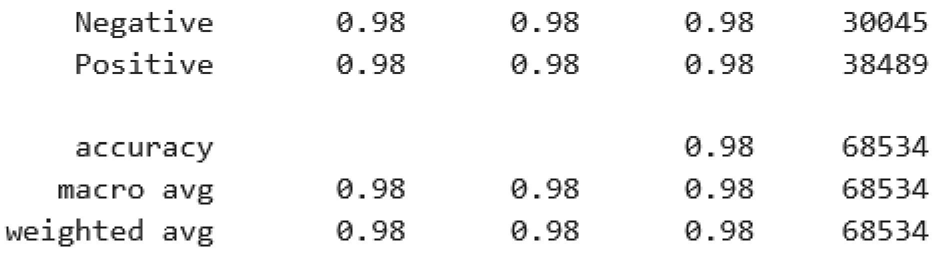


Figure 4 BERT Training Confusion matrix

- **Classification report:**



```
    Negative       0.98      0.98      0.98     30045
    Positive       0.98      0.98      0.98     38489

    accuracy                           0.98     68534
   macro avg       0.98      0.98      0.98     68534
weighted avg       0.98      0.98      0.98     68534
```

4.2.2 Testing F1 Score and Confusion Matrix

The model was then evaluated on the testing data to determine its generalization performance.

- **Testing F1 Score**: 0.891
- **Testing Confusion Matrix**:

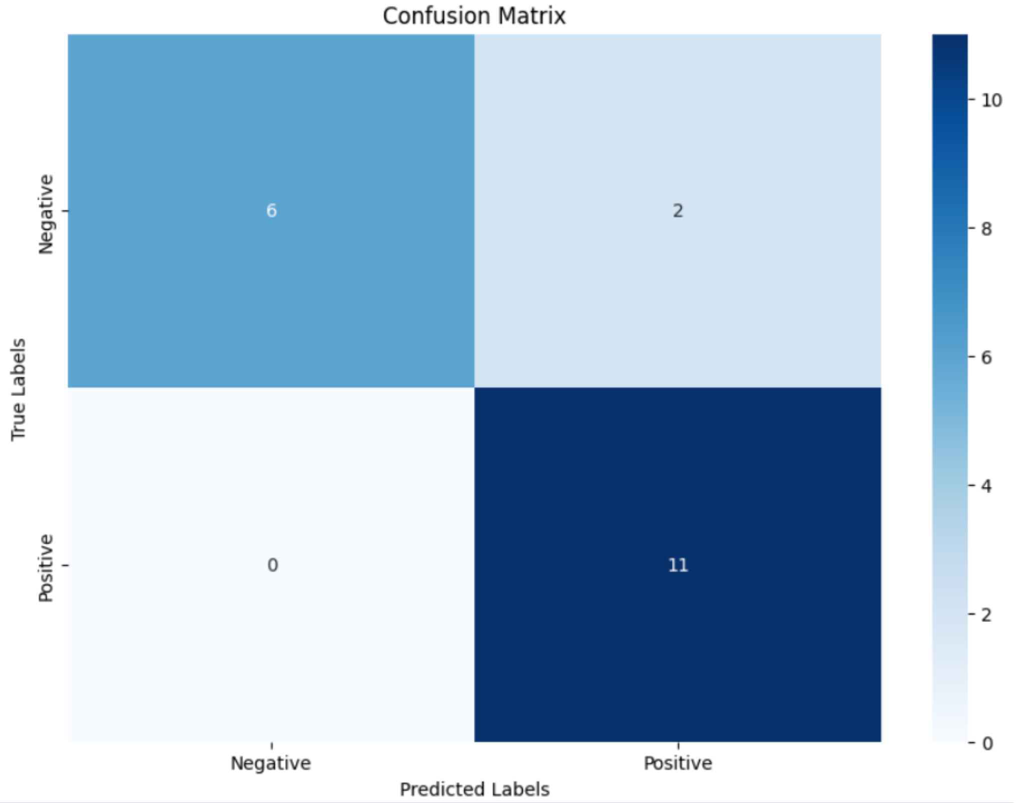


Figure 5 BERT Testing Confusion Matrix

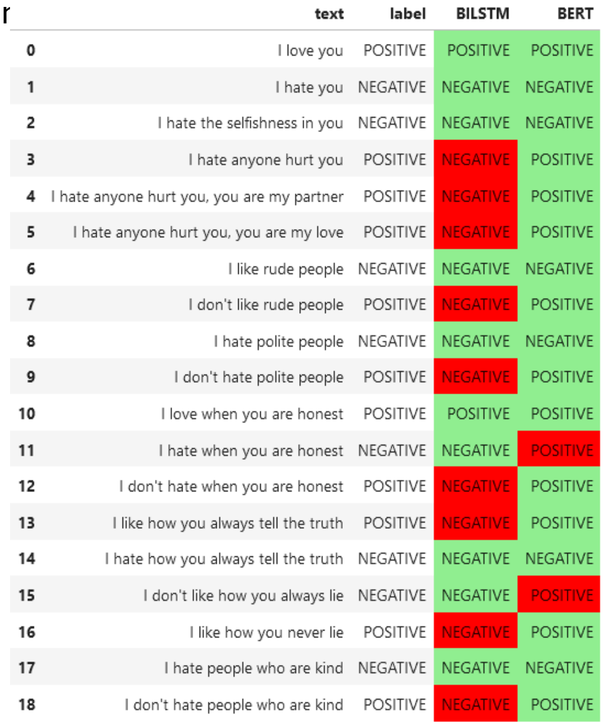
4.3 Results and Conclusion

The results indicate that the BERT model addresses several issues observed with the BiLSTM model:

1. **Context Understanding**: BERT effectively resolves the problem of context sensitivity that BiLSTM struggled with. For example, in the sentence “I hate anyone hurt you,” BERT correctly labels the sentiment as positive, whereas BiLSTM misclassifies it as negative. This demonstrates BERT’s superior ability to capture nuanced sentiment through context-aware embeddings.

2. **Handling Long Sequences**: The BERT model also improves upon handling long sequence lengths. In the example "I hate anyone hurt you, you are my partner," BERT correctly identifies the overall positive sentiment, whereas BiLSTM fails to properly integrate the context at the end of the sentence. This illustrates BERT's enhanced capability to consider the entire sequence for accurate sentiment analysis.
3. **Contextual Negation**: BERT performs better at understanding context in negations compared to BiLSTM. For instance, in "I don't like rude people," BERT correctly interprets the positive sentiment of the context, whereas BiLSTM misclassifies it by focusing too narrowly on the negative words "don't" and "rude."

These results highlight BERT's improved performance in capturing complex contextual information and handling longer sequences, addressing the limitations observed with BiLSTM r



	text	label	BILSTM	BERT
0	I love you	POSITIVE	POSITIVE	POSITIVE
1	I hate you	NEGATIVE	NEGATIVE	NEGATIVE
2	I hate the selfishness in you	NEGATIVE	NEGATIVE	NEGATIVE
3	I hate anyone hurt you	POSITIVE	NEGATIVE	POSITIVE
4	I hate anyone hurt you, you are my partner	POSITIVE	NEGATIVE	POSITIVE
5	I hate anyone hurt you, you are my love	POSITIVE	NEGATIVE	POSITIVE
6	I like rude people	NEGATIVE	NEGATIVE	NEGATIVE
7	I don't like rude people	POSITIVE	NEGATIVE	POSITIVE
8	I hate polite people	NEGATIVE	NEGATIVE	NEGATIVE
9	I don't hate polite people	POSITIVE	NEGATIVE	POSITIVE
10	I love when you are honest	POSITIVE	POSITIVE	POSITIVE
11	I hate when you are honest	NEGATIVE	NEGATIVE	POSITIVE
12	I don't hate when you are honest	POSITIVE	NEGATIVE	POSITIVE
13	I like how you always tell the truth	POSITIVE	NEGATIVE	POSITIVE
14	I hate how you always tell the truth	NEGATIVE	NEGATIVE	NEGATIVE
15	I don't like how you always lie	NEGATIVE	NEGATIVE	POSITIVE
16	I like how you never lie	POSITIVE	NEGATIVE	POSITIVE
17	I hate people who are kind	NEGATIVE	NEGATIVE	NEGATIVE
18	I don't hate people who are kind	POSITIVE	NEGATIVE	POSITIVE

Figure 6 BERT Result Table

5.0 Overall Conclusion

The comparison between the BiLSTM and BERT models reveals significant improvements in sentiment analysis capabilities when using BERT:

1. **Enhanced Context Sensitivity**: BERT demonstrates a superior ability to understand and interpret nuanced contexts, overcoming the limitations faced by BiLSTM. This is evident in its correct classification of sentences like “I hate anyone hurt you” as positive, which BiLSTM misclassified.
2. **Effective Handling of Long Sequences**: BERT excels in processing and accurately interpreting long sentences. It successfully captures the overall sentiment even when the critical context appears towards the end of a sentence, as shown in the example “I hate anyone hurt you, you are my partner,” which BERT correctly labels as positive.
3. **Improved Negation Handling**: BERT's contextual embeddings allow it to better understand sentences with negations. It correctly identifies the positive sentiment in sentences like “I don’t like rude people,” whereas BiLSTM incorrectly labels such sentences as negative by focusing on individual negative words.
4. **Context Embeddings vs. LSTM Embeddings**: The key difference lies in how context is embedded. LSTM embeddings are sequential and depend heavily on the order of words, often failing to capture long-range dependencies effectively. In contrast, BERT utilizes bidirectional transformer architecture, allowing it to consider the full context of a word in both directions, resulting in richer and more accurate context embeddings. This difference enables BERT to handle complex sentence structures and contextual information more effectively than LSTM models.

Overall, BERT provides a more robust and accurate approach to sentiment analysis, effectively addressing the challenges of context sensitivity, sequence length, and negation that BiLSTM models struggle with. This makes BERT a preferable choice for applications requiring nuanced sentiment interpretation and complex contextual understanding.

6.0 Tools and Resources

6.1 Tools

- **Python**: The primary programming language used for developing the entire pipeline.
- **Jupyter Notebook**: For interactive development, visualization, and documentation of the code.
- **PyTorch**: Used to build, train, and evaluate neural network models such as LSTM and BERT.
- **Transformers**: For pre-trained BERT model and tokenization.
- **NLTK**: For natural language processing tasks like tokenization, stopword removal, and stemming.
- **Pandas**: For data manipulation and analysis.
- **NumPy**: For numerical operations and handling arrays.
- **scikit-learn**: For machine learning utilities, including the calculation of performance metrics such as the F1 score and splitting data into train/test sets.
- **torchtext**: For using pre-trained GloVe embeddings.
- **Matplotlib/Seaborn**: For data visualization to create plots for analysis and comparison of model performance.
- **tqdm**: For displaying progress bars during data processing and model training.
- **Kaggle:**For training and testing models

6.2 Resources

1. **Research Papers and Articles**:
    - **A Survey on Contextual Embeddings**: A comprehensive overview of contextual embeddings and their applications in NLP.
    - **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding** by Jacob Devlin et al.: Foundational knowledge about BERT and its applications.
    - **Attention Is All You Need**by Vaswani et al.: The original paper introducing the Transformer model, which BERT is based on.

2. **Online Articles and Tutorials**:
    - **Medium Article: Contextual vs. Non-Contextual Word Embedding Models for Hindi Named Entity Recognition**: Link
3. **Datasets**:
    - **Stanford Sentiment Treebank (SST-2)**: A dataset specifically designed for sentiment analysis tasks, available via StanfordNLP.

7.0 Biggest Challenges Faced

1. Creating a Custom Dataset:

- Challenge: Existing datasets were insufficient for capturing the nuanced context of homonyms in sentiment analysis.
- Solution: You created a custom dataset with specific examples of homonyms to ensure the model could learn to handle these cases effectively.

2. Fine-Tuning Models:

- Challenge: Achieving optimal performance on the sentiment analysis task required extensive fine-tuning of various models.
- Solution: You experimented with and fine-tuned several models, including BERT and AlexNet, on sentiment analysis tasks to identify the best-performing model for your dataset.

3. Model Selection and Evaluation:

- Challenge: Identifying the most suitable model for handling homonyms and achieving high accuracy in sentiment classification.
- Solution: By trying out different models and fine-tuning them specifically for the sentiment analysis task, you were able to evaluate their performance and choose the best one for your needs.

4. Data Preparation and Augmentation:

- Challenge: Preparing and augmenting the dataset to ensure it was robust enough for training the models.
- Solution: Implementing thorough preprocessing steps and augmenting the dataset as necessary to improve model training and performance.

8.0 Lessons Learned

1. **Importance of Custom Datasets**:
    - **Lesson**: Generic datasets often fail to capture specific nuances required for specialized tasks such as homonym sentiment analysis.

- **Takeaway**: Creating a custom dataset tailored to the specific problem at hand can significantly improve model performance and ensure better handling of unique cases.

2. **Model Flexibility and Experimentation**:
   - **Lesson**: Different models have varying strengths and weaknesses. It’s essential to experiment with multiple models to find the best fit for your specific task.
   - **Takeaway**: Fine-tuning and evaluating a variety of models, such as BERT and AlexNet, is crucial for achieving optimal results in sentiment analysis tasks.
3. **Fine-Tuning for Specific Tasks**:
   - **Lesson**: Pre-trained models need to be fine-tuned on your specific dataset to perform well on the target task.
   - **Takeaway**: Invest time in fine-tuning pre-trained models on your dataset to adapt them to the specific nuances and requirements of your sentiment analysis task.
4. **Comprehensive Data Preparation**:
   - **Lesson**: Proper data preparation and augmentation are fundamental to improving model training and performance.
   - **Takeaway**: Ensure thorough preprocessing and consider data augmentation techniques to enhance the quality and robustness of your training dataset.
5. **Handling Long Sentences and Context**:
   - **Lesson**: Some models, like BiLSTM, may struggle with long sentences and complex contexts.
   - **Takeaway**: Using models like BERT, which are better at handling long sequences and capturing context, can lead to improved sentiment classification accuracy.
6. **Understanding Contextual Embeddings**:
   - **Lesson**: Contextual embeddings provided by models like BERT are more effective in capturing the meaning of words in different contexts compared to traditional embeddings.
   - **Takeaway**: Leveraging contextual embeddings is critical for tasks that require nuanced understanding of language, such as sentiment analysis involving homonyms.