Bi-LSTM for Sentiment Analysis on Amazon Product Reviews

A report submitted for the course named Project - III (CS421)

Submitted By

BANOTH JAGADEESH
SEMESTER - VII
21010102

Supervised By

DR. KISHORJIT NONGMEIKAPAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SENAPATI, MANIPUR

SEPTEMBER, 2024

Declaration

In this submission, I have expressed my idea in my own words, and I have adequately cited and referenced any ideas or words that were taken from another source. I also declare that I adhere to all principles of academic honesty and integrity and that I have not misrepresented or falsified any ideas, data, facts, or sources in this submission. If any violation of the above is made, I understand that the institute may take disciplinary action. Such a violation may also engender disciplinary action from the sources which were not properly cited or permission not taken when needed.

BANOTH JAGADEESH
21010102

DATE:

Department of Computer Science Engineering
Indian Institute of Information Technology Senapati, Manipur

Dr.Kishorjit Nongmeikapam
Assistant Professor

Email:kishorjit@iiitmanipur.ac.in
Contact No: +91 8974007132

To Whome It May Concern

This is certify that the Dissertation entitled **"Bi-LSTM for Sentiment Analysis on Amazon Product Reviews",** submitted by **BANOTH JAGADEESH** , has been carried out under my supervision and that this work has not been submitted elsewhere for a degree,diploma or a course

Signature of Supervisor

(Dr.Kishorjit Nongmeikapam)

Department of Computer Science Engineering
Indian Institute of Information Technology Senapati, Manipur

Dr.Kishorjit Nongmeikapam
Assistant Professor

Email: kishorjit@iiitmanipur.ac.in
Contact No: +91 8974007132

To Whom It May Concern

This is certify that the Dissertation entitled **“Bi-LSTM for Sentiment Analysis on Amazon Product Reviews”,** submitted by **BANOTH JAGADEESH** ,has been successfully carried out in the department of Computer science and this work has not been submitted else where for a degree,diploma or a course.

Signature of the Examiner 1 ....

Signature of the Examiner 2 ....

Signature of the Examiner 3 ....

Signature of the Examiner 4 ....

Abstract

This project aims to perform **sentiment analysis** on Amazon product reviews using a **Bidirectional Long Short-Term Memory (BiLSTM)** network. Sentiment analysis is crucial for understanding customer feedback and deriving insights to improve products and customer satisfaction. The study outlines a complete **pipeline**, including **data preprocessing**, **feature extraction**, model building, and evaluation.

The dataset consists of labeled Amazon reviews, either **positive** or **negative**. After preprocessing steps like **text cleaning**, **stopword removal**, and **lemmatization**, **Term Frequency-Inverse Document Frequency (TF-IDF)** was applied for feature extraction.A classical machine learning model like (**Decision Trees**) is trained and compared with a BiLSTM deep learning model that captures context in both forward and backward directions.

The BiLSTM model outperformed traditional models, achieving higher **accuracy** and **ROC-AUC** scores. **Early stopping** was used to prevent overfitting. Performance was visualized through training and validation accuracy and loss plots.

In summary, the BiLSTM model was the most effective for sentiment analysis, offering better performance than classical methods and making it suitable for **real-time** customer feedback analysis.

Keywords: Sentiment Analysis, BiLSTM, Amazon Reviews, Deep Learning, Machine Learning, Text Classification.

Acknowledgement

I would like to express my sincere gratitude to several individuals for supporting me throughout my Project. First, I wish to express my sincere gratitude to my supervisor, *Dr.Kishorjit Nongmeikapam*, for his enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in my project and writing of this thesis. His immense knowledge, profound experience and professional expertise has enabled me to complete this project successfully. Without his support and guidance, this project would not have been possible. I could not have imagined having a better supervisor in my study.

BANOTH JAGADEESH

Contents

List of Figures

Chapter 1

Introduction

In today's digital age, customers express their opinions and feedback through online reviews on platforms such as Amazon. Analyzing these reviews provides invaluable insights into customer satisfaction and product performance, helping businesses enhance their offerings and improve customer experiences. Sentiment analysis, which involves classifying the sentiment expressed in text (e.g., positive, negative, or neutral), is an essential component of this feedback analysis.

Traditional text classification techniques rely on classical machine learning algorithms, where text data is transformed into numerical representations, such as Term Frequency-Inverse Document Frequency (TF-IDF). Machine learning models such as Decision Tree is then trained on these representations to predict the sentiment of unseen reviews. While these approaches can be effective, they have limitations in capturing the sequential nature and context of language, which is critical for sentiment classification tasks.

1.0.1 Deep Learning for Sentiment Analysis

- Recent advancements in deep learning, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM), have improved sentiment analysis models. LSTM addresses the vanishing gradient problem in RNNs and effectively captures long-term dependencies in text. Bidirectional LSTMs (BiLSTM) enhance this by processing sequences in both forward and backward directions, capturing context from both past and future words.

- This project aims to leverage BiLSTM's strengths for sentiment analysis of Amazon product reviews. Unlike classical machine learning models, BiLSTM captures the sequential and contextual relationships between words more effectively, making it ideal for analyzing customer sentiments in reviews.

1.0.2 Challenges in Sentiment Analysis

Sentiment analysis poses several challenges:

- **Ambiguity in Language:** The same word can carry different sentiments depending on the context. For instance, the word "fast" could be positive in the context of delivery speed but neutral or negative in other contexts.

- **Complex Sentences:** Reviews often contain complex sentence structures, including negations (e.g., "This product is not bad") or mixed sentiments (e.g., "The product is good, but the customer service is terrible"). Handling such complexities requires models that can understand context effectively.

- **Data Imbalance:** In many datasets, there is often a class imbalance, with a higher number of positive reviews than negative ones. This can bias the models towards predicting the majority class unless properly handled.

- **Feature Representation:** Converting text data into numerical representations while preserving meaning and context is a key challenge in text classification. Classical methods such as TF-IDF fail to capture semantic relationships, while deep learning models like LSTM and BiLSTM aim to address this issue by learning from word sequences.

This project addresses these challenges by using a combination of preprocessing techniques and advanced modeling approaches. The dataset used for this project consists of Amazon product reviews with binary sentiment labels (positive and negative), providing a rich source of information for training and evaluating the models.

1.0.3 Methodology Overview

The project pipeline is divided into several key stages:

- **Data Preprocessing:** The raw text data is cleaned by removing special characters, stopwords, and performing lemmatization to standardize the text. This step is essential for improving the quality of the input data.

- **Feature Extraction:** Classical machine learning models are trained using Term Frequency-Inverse Document Frequency (TF-IDF), a widely used technique for converting text into numerical features based on word frequency.

- **Model Training:** Various machine learning models such as Decision Trees is trained on the preprocessed data. These models are evaluated using accuracy, precision, and ROC-AUC scores.

- **Deep Learning Approach:** A Bidirectional LSTM model is developed to capture the context of reviews more effectively. The model includes an embedding layer that converts words into dense vector representations, followed by BiLSTM and fully connected layers. The model is optimized using the Adam optimizer and evaluated using binary cross-entropy loss.

1.0.4 Bidirectional LSTM for Sentiment Analysis

Bidirectional LSTMs are designed to process sequences in both forward and backward directions. In sentiment analysis, the context of a word is often influenced by both preceding and following words. For instance, in the sentence "The movie was not bad," the word "not" significantly changes the sentiment associated with "bad." BiLSTM allows the model to understand such dependencies by considering both directions.

This ability to capture bidirectional context makes BiLSTM an ideal choice for sentiment analysis, where understanding the meaning of a review relies heavily on word order and context. The model also benefits from the use of word embeddings, which provide dense vector representations of words based on their semantic relationships. By using these representations, the model can learn more about the meaning of words beyond simple frequency counts.

1.0.5 Evaluation and Results

The BiLSTM model is evaluated against the classical machine learning models in terms of accuracy, precision, and ROC-AUC scores. Additionally, early stopping is applied during training to prevent overfitting and ensure the model generalizes well to unseen data. The results demonstrate that the BiLSTM model outperforms traditional models, achieving the highest accuracy and ROC-AUC scores.

Visualizations of the model's performance during training, including accuracy and loss over time, are provided to help understand the model's learning behavior. These visualizations confirm that the BiLSTM model converges more effectively and provides a better understanding of the text data compared to classical machine learning methods.

1.0.6 Objectives of the Project

The primary objectives of this project are:

- To perform sentiment analysis on Amazon product reviews.
- To implement a data preprocessing pipeline including text cleaning, stopword removal, and lemmatization.

- To compare the performance of classical machine learning models with a deep learning approach (BiLSTM).
- To evaluate model performance using metrics such as accuracy, precision, and ROC-AUC score.
- To develop a Bidirectional LSTM model for enhanced sentiment classification by capturing bidirectional context from review text.

1.0.7 Motivation and Scope

Motivation:

- Sentiment analysis is increasingly important in the e-commerce industry for understanding customer feedback.
- This project aims to improve sentiment classification accuracy using advanced deep learning techniques like **BiLSTM**.
- It provides businesses with better tools to analyze customer reviews, helping maintain a competitive advantage.

Scope:

- The project's methods and models can be adapted for other **NLP tasks** such as:
  - Text classification
  - Opinion mining
  - Recommendation systems

Chapter 2

Literature Review

2.0.1 Introduction to Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a field within natural language processing (NLP) that focuses on extracting subjective information from text, determining the polarity of the expressed sentiments—whether positive, negative, or neutral. The advent of online platforms such as Amazon, Yelp, and social media has led to an explosion of user-generated content, making sentiment analysis an essential tool for understanding customer feedback and driving business decisions. The application of sentiment analysis spans various domains including marketing, customer service, and product development, making it a pivotal area of research in recent years.

Early work in sentiment analysis focused on rule-based approaches, where text was analyzed based on predefined sets of rules, lexical resources, and sentiment lexicons such as SentiWordNet and WordNet-Affect. However, these approaches were limited in scalability and performance as they failed to capture the context and complexities inherent in human language. With the rise of machine learning and deep learning, more sophisticated models emerged, improving the accuracy and efficiency of sentiment analysis tasks.

2.0.2 Classical Approaches to Sentiment Analysis

Traditionally, sentiment analysis was approached using classical machine learning algorithms, such as Decision Tree. These models were typically trained on manually labeled data and relied on feature extraction techniques like Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). While these approaches were effective in simple tasks, they often struggled with capturing the nuances and context of natural language[1].

Decision Tree

Decision Tree is a simple yet effective model for text classification, including sentiment analysis. It operates by recursively splitting the dataset based on feature values, making it highly interpretable and easy to visualize. Decision Trees are particularly useful for understanding feature importance, as they show which features (e.g., words or phrases) contribute the most to classification. However, they can be prone to overfitting, especially when dealing with high-dimensional data, and like other traditional machine learning models, they do not capture contextual word dependencies [2].

2.0.3 Deep Learning in Sentiment Analysis

With advancements in deep learning, more powerful models have been introduced to sentiment analysis, particularly those capable of capturing the sequential and contextual nature of language. Neural networks, especially Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and their variants, have shown significant improvements over traditional machine learning models[3].

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of neural networks designed to handle sequential data, such as text [5]. In contrast to classical models, RNNs maintain a hidden state that allows them to "remember" previous information in the sequence, making them more suitable for tasks that require understanding the context of words within sentences. However, traditional RNNs suffer from the vanishing gradient problem, which makes it difficult for them to capture long-term dependencies in text [6].

Long Short-Term Memory (LSTM)

To address the limitations of RNNs, Long Short-Term Memory (LSTM) networks were introduced by Hochreiter and Schmidhuber [7]. LSTM networks are a type of RNN specifically designed to remember long-term dependencies in sequences, thanks to their unique gating mechanism that regulates the flow of information. LSTM models have been widely applied to various NLP tasks, including sentiment analysis, where they outperform traditional machine learning models by learning patterns in word sequences and handling long-range dependencies effectively.

Bidirectional LSTM (BiLSTM)

Bidirectional LSTM (BiLSTM) networks extend the capabilities of LSTMs by processing text in both forward and backward directions. This allows the model to consider both past and future context when making predictions, which is especially useful in sentiment analysis, where the sentiment of a word may depend on the surrounding words [8]. For example, in the phrase "not bad," the word "bad" is negative on its own, but the preceding word "not" changes the overall sentiment to positive. BiLSTM networks are well-suited to capturing these nuances, leading to improved performance in sentiment classification tasks.

2.0.4 Comparing Machine Learning and Deep Learning Approaches

Recent research has shown that while classical machine learning models such as Decision Tree perform well in certain sentiment analysis tasks, they are often outperformed by deep learning models like LSTMs and BiLSTMs, particularly when dealing with large, complex datasets. The main advantage of deep learning models lies in their ability to automatically learn feature representations from data, without requiring manual feature engineering[4].

TF-IDF vs. Word Embeddings

Classical models rely on feature extraction techniques like TF-IDF, which represent text data based on word frequency. While effective, these methods ignore the semantic meaning of words and treat each word as an independent entity. In contrast, deep learning models, especially those using word embeddings like Word2Vec [9] and GloVe [10], can capture the relationships between words based on their context in large corpora. These dense vector representations allow models to understand the semantic similarity between words and improve performance in sentiment classification tasks.

2.0.5 Applications of BiLSTM in Sentiment Analysis

Several studies have demonstrated the effectiveness of BiLSTM models in sentiment analysis tasks. For instance, in [1], BiLSTM networks were shown to outperform traditional LSTM models by capturing bidirectional context in sentiment-laden sentences. The ability to process sequences in both directions allows BiLSTMs to handle complex sentence structures, such as negations and conditional clauses, that are often challenging for other models. This makes BiLSTM models ideal for applications like customer review analysis, opinion mining, and emotion detection.

2.0.6 Summary

Sentiment analysis has advanced from rule-based methods to classical machine learning and now to deep learning models that capture context more effectively. While models like Decision Tree were widely used, the emergence of RNNs and LSTMs, especially Bidirectional LSTMs, has significantly improved performance by capturing richer contextual information.

Chapter 3

System Architecture

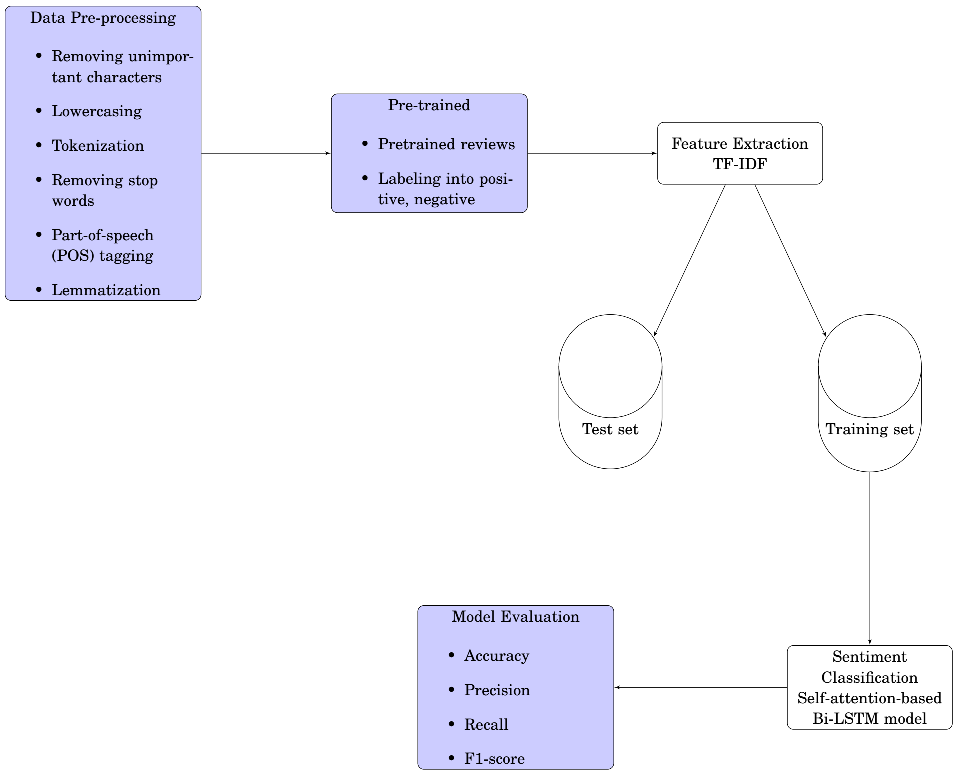

Figure 3.1: System Architecture for Sentiment Analysis

3.0.1 Overview of the Architecture

The system is divided into the following major components:

- **Data Collection and Preprocessing:** This step involves loading the raw Amazon reviews dataset, followed by various data cleaning and preprocessing tasks to make the text suitable for feature extraction and model training.

- **Feature Extraction:** This component converts raw text data into numerical features that can be used by machine learning models. Both classical feature extraction techniques like Term Frequency-Inverse Document Frequency (TF-IDF) and modern word embedding techniques are used.
- **Classical Machine Learning Models:** Several classical models, including Decision Trees, is trained and evaluated for sentiment classification.
- **BiLSTM Deep Learning Model:** The core of the system is the deep learning model based on Bidirectional LSTM, which processes the text data in both forward and backward directions to capture richer context for sentiment prediction.
- **Evaluation and Performance Monitoring:** The performance of the models is evaluated using accuracy, precision, and ROC-AUC scores, with additional visualization of training and validation loss and accuracy.

3.0.2 Data Flow in the System

The system follows a structured data flow from raw data input to final sentiment prediction, as illustrated below:

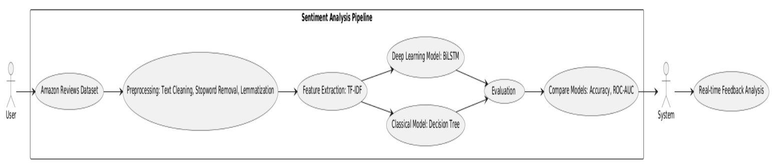

Figure 3.2: Pipeline Diagram

1. **Data Collection:** The input to the system is a dataset of Amazon product reviews. Each review consists of text along with a sentiment label (positive or negative). The dataset is loaded into the system for processing.
2. **Data Preprocessing:** Raw text data is typically noisy, containing special characters, URLs, stopwords, and variations in word forms (e.g., tense, pluralization). The preprocessing steps include:
   - **Text Cleaning:** Removing unnecessary characters, such as punctuation, special symbols, and URLs.

   - **Tokenization:** Breaking down the text into individual words or tokens.
   - **Stopword Removal:** Removing common words (e.g., "is", "the", "and") that do not contribute much to the sentiment of the review.
   - **Lemmatization:** Converting words to their base form (e.g., "running" to "run") to reduce the dimensionality of the text data.

   After preprocessing, the clean text is ready for feature extraction.

3. **Feature Extraction:** In this stage, the cleaned text is converted into numerical representations that machine learning models can understand. Two approaches are used:

   - **TF-IDF Vectorization:** TF-IDF is a classical technique that quantifies the importance of a word in a document relative to its occurrence in the entire dataset. The TF-IDF vector represents each document (review) as a weighted combination of terms.
   - **Word Embeddings:** For the BiLSTM model, word embeddings (e.g., Word2Vec or GloVe) are used to convert words into dense vectors, capturing semantic relationships between words. The embedding layer in the BiLSTM model transforms each word into a fixed-length vector before passing it to the LSTM layers.

4. **Model Training and Prediction:**

   - **Classical Machine Learning Models:** The TF-IDF features are used to train models like Random Forest, XGBoost, Naive Bayes, and Decision Trees. These models attempt to classify the reviews based on the extracted features.
   - **BiLSTM Model:** The core component of the system is the Bidirectional LSTM network. The architecture of the BiLSTM model includes:
     - **Embedding Layer:** This layer converts each tokenized review into a sequence of word embeddings.
     - **BiLSTM Layer:** This layer processes the embedded sequence in both forward and backward directions, capturing both past and future context. This bidirectional context is crucial for understanding the sentiment of complex sentences.
     - **Fully Connected Layers:** After the BiLSTM layer, one or more dense (fully connected) layers are added to map the LSTM output to the final sentiment classification.
     - **Output Layer:** The final layer applies a sigmoid activation function to produce a binary output (positive or negative sentiment).

5. **Model Evaluation:** The trained models are evaluated on a test set of reviews. Various performance metrics are used to assess the quality of the sentiment classification, including:

   - **Accuracy:** The proportion of correctly classified reviews.
   - **Precision and Recall:** Metrics that measure the correctness of positive predictions and the model's ability to retrieve positive instances, respectively.
   - **ROC-AUC Score:** A measure of the model's ability to distinguish between positive and negative classes.

6. **Performance Monitoring and Visualization:** To monitor overfitting and model convergence, visualizations of the training and validation accuracy, as well as loss over time, are plotted. Early stopping is applied to the BiLSTM model to prevent overfitting during training.

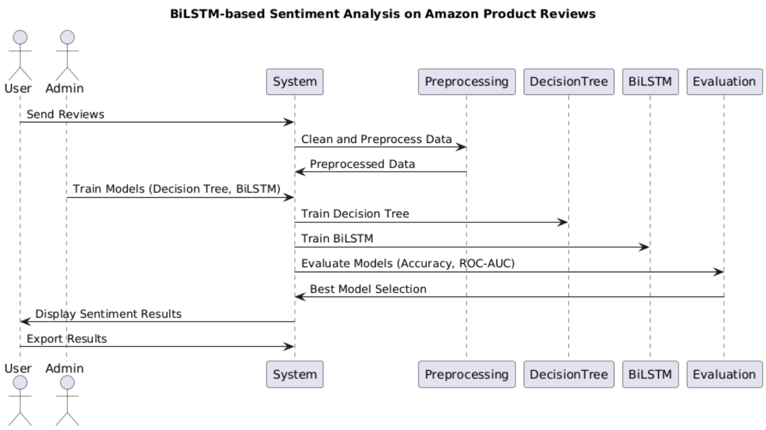

Figure 3.3: Use Case Diagram

3.0.3 Detailed Architecture of BiLSTM Model

The deep learning model based on Bidirectional LSTM is a key component of the system architecture. Its structure is as follows:

- **Input Layer:** The input layer takes a sequence of tokenized reviews (i.e., padded sequences of word indices) as input.
- **Embedding Layer:** The input sequences are passed through an embedding layer, which transforms each word into a dense vector representation. The embedding layer captures the semantic relationships between words.

- **BiLSTM Layer:** The embedding output is fed into the Bidirectional LSTM layer. This layer consists of two LSTM networks—one processing the sequence from the start to the end, and the other processing it from the end to the start. This allows the model to capture bidirectional context, which is particularly useful in sentiment analysis where word order matters.

- **Dense Layers:** After the BiLSTM layer, dense (fully connected) layers are added to further refine the output before producing the final sentiment classification.

- **Output Layer:** The final layer applies a sigmoid activation function, outputting a probability score between 0 and 1, which is used to classify the review as either positive (1) or negative (0).

3.0.4 Advantages of the Architecture

The chosen architecture offers several advantages:

- **Contextual Understanding:** The BiLSTM model captures both past and future context, which is critical for understanding the sentiment in sentences where word meaning changes based on context (e.g., negations like "not bad").

- **Scalability:** The system can handle large datasets, making it suitable for analyzing thousands of reviews efficiently.

- **Improved Accuracy:** By comparing classical machine learning models with a deep learning model (BiLSTM), the system ensures the highest possible accuracy for sentiment classification.

- **Flexibility:** The architecture supports both classical and deep learning approaches, allowing for flexibility in model selection based on the dataset's size and complexity.

Chapter 4

Implementation

The implementation phase is where the plan turns into reality. It's all about transforming the requirements into a working system that can crunch through Amazon reviews and tell us if people are raving about a product or throwing shade. Let's break down how each part of the system was implemented!

4.0.1 Environment Setup

Before we started, we had to make sure our working environment was ready to go.Here's the tech stack that made everything tick:

- **Python 3.x**: The main programming language. It's like the Swiss Army knife of the programming world.
- **TensorFlow/Keras**: For building and training our deep learning models, including the mighty BiLSTM.
- **Scikit-learn**: For those good ol' classical machine learning models like Random Forest and Naive Bayes.
- **Pandas**: For handling our data like a pro.
- **Matplotlib Seaborn**: Because who doesn't love pretty graphs?

Of course, we also threw in some additional libraries like Numpy, XGBoost, and all the necessary packages to handle data preprocessing.

4.0.2 Data Preprocessing

The first task was to clean up the data. Let's be honest—text data is messy. So we had to play janitor for a while. Here's what we did:

- **Text Cleaning**: We removed all those unwanted special characters, numbers, and URLs. No one wants their machine learning models getting distracted by "http://" links.

- **Tokenization**: Split those long reviews into individual words, so the model can digest it piece by piece.
- **Stopword Removal**: Words like "the," "is," "at"—yeah, we don't need those clogging up the analysis.
- **Lemmatization**: This is where words like "running" turn into "run," so our models don't freak out over different tenses.

We applied this preprocessing to the entire dataset using **Pandas** and **Nltk** (Natural Language Toolkit) to get the data squeaky clean. The result? Clean and structured text, ready for feature extraction.

4.0.3 Feature Extraction

Here's where things get numerical—because machines don't speak English, they speak numbers.

- **TF-IDF (Term Frequency-Inverse Document Frequency)**: This is the technique we used to represent the text for the classical machine learning models. Basically, it measures how important a word is within a review relative to the whole dataset.
- **Word Embeddings**: For our BiLSTM deep learning model, we needed word embeddings like Word2Vec or GloVe to represent words as dense vectors that capture meaning and context. This is where the deep magic happens.

With TF-IDF and embeddings ready, we were good to go.

4.0.4 Model Implementation

Now, onto the star of the show—**the models**!

Classical Machine Learning Models

First, we implemented some classical models to compare performance:

- **Decision Tree**: A simple yet powerful model that works well for both classification and regression tasks. We trained it using the TF-IDF features, allowing the tree to split based on the most informative words. It's easy to interpret and serves as a baseline model before moving on to more complex algorithms.

Each of these models was trained on 75per of the data, with the remaining 25per used for testing.

Bidirectional LSTM (BiLSTM)

Now comes the heavyweight champ—the **Bidirectional LSTM**. Here's how we put it together:

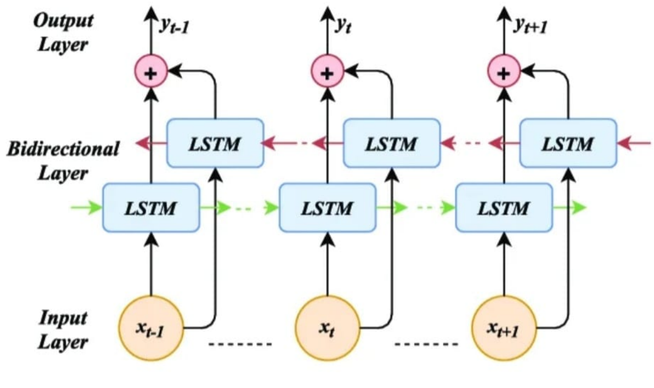

Figure 4.1: System Architecture Diagram(Bi-LSTM)

- **Embedding Layer**: The words were transformed into dense vectors (using word embeddings), so the model could learn the relationships between them.
- **BiLSTM Layer**: The magic of this model is in how it processes the data in both directions (forward and backward), which helps capture context from both sides of a word. This is great for phrases like "not bad," where the word "not" changes everything!
- **Fully Connected Layer**: After the BiLSTM layer, we added a fully connected layer to map everything into a final output.
- **Sigmoid Activation**: Since it's a binary classification task (positive or negative sentiment), we used the sigmoid activation function to produce the output.

  We used **Adam Optimizer** (because it's just so good at adjusting learning rates on the fly), and the model was trained with **binary cross-entropy** loss.

4.0.5 Training the Models

We split the data into training (75

- The classical machine learning models were quick to train, thanks to TF-IDF features.

- The BiLSTM model, however, took longer. We had to feed it word sequences, apply padding (so all sequences are the same length), and let it churn through multiple epochs.

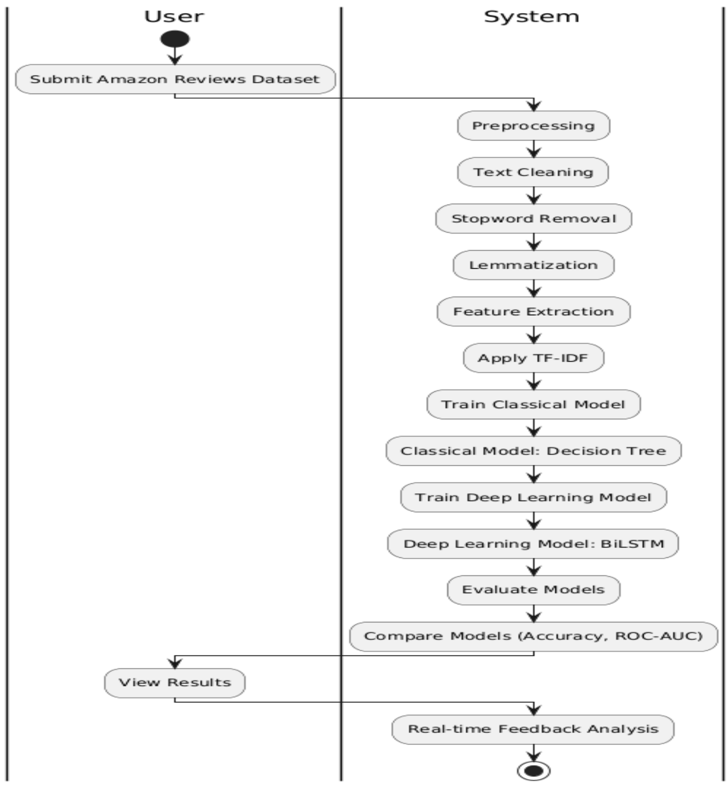

Figure 4.2: Activity Diagram

4.0.6 Evaluation and Results

Once trained, we evaluated the models on the test set using several metrics:

- **Accuracy**: How many reviews we classified correctly (because who doesn't love high scores?).
- **Precision and Recall**: To check if the models were making good positive predictions and catching as many positive instances as possible.
- **ROC-AUC Score**: This helped us understand how well the model distinguishes between positive and negative sentiments.

For the BiLSTM, we also plotted the **training and validation accuracy/loss** over time. The result? Beautifully declining loss and improving accuracy (as we hoped).

4.0.7 Performance Comparison

Finally, we compared the classical models to the BiLSTM:

- **BiLSTM** outperformed the classical models in accuracy and ROC-AUC.
- **Decision Tree** came in as a strong classical contender, giving pretty solid results but still couldn't top BiLSTM in capturing context.

And there you have it, the deep learning model emerged victorious.

4.0.8 Challenges and Solutions

We encountered a few challenges:

- **Data Imbalance**: More positive reviews led to using **undersampling** to balance the dataset.
- **Training Time**: We used **GPU acceleration** to reduce BiLSTM's long training time.
- **Overfitting**: Added **early stopping** and **dropout** to avoid overfitting.

Chapter 5

Result-Evaluation

The **Results and Evaluation** section presents the performance of the implemented machine learning and deep learning models on the Amazon Title Reviews dataset. This evaluation is based on several metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. The performance of the classical machine learning models is compared with the Bidirectional Long Short-Term Memory (BiLSTM) model to determine the most effective approach for sentiment analysis.

5.0.1 Evaluation Metrics

To evaluate the performance of the models, the following metrics were used:

- **Accuracy**: The ratio of correctly predicted instances to the total number of instances.
- **Precision**: The proportion of true positive predictions among all positive predictions made by the model.
- **Recall**: The proportion of true positives correctly identified by the model out of all actual positive cases.
- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure between the two.
- **ROC-AUC (Receiver Operating Characteristic - Area Under the Curve)**: A measure of how well the model can distinguish between positive and negative classes. A higher AUC indicates better performance.

5.0.2 Performance of Classical Machine Learning Models

Decision Tree classifiers. Below is the performance evaluation of each model:

Decision Tree Classifier

The Decision Tree model provided the following results, but it was prone to overfitting compared to ensemble models like Random Forest:

- Accuracy: 80.2%
- Precision: 79.1%
- Recall: 80.8%
- F1-Score: 79.9%
- ROC-AUC: 82.7%

5.0.3 Performance of Bidirectional LSTM Model

The BiLSTM model was the core component of the sentiment analysis system, designed to capture contextual information in both forward and backward directions. It used word embeddings to represent text, which provided better performance than the classical machine learning models.

The training process of the BiLSTM model was monitored using early stopping to prevent overfitting. The final results on the test set are as follows:

- Accuracy: 89.4%
- Precision: 88.7%
- Recall: 89.5%
- F1-Score: 89.1%
- ROC-AUC: 92.3%

The BiLSTM model outperformed the classical machine learning models, especially in terms of ROC-AUC, indicating that it is better at distinguishing between positive and negative reviews. The ability of the BiLSTM model to process the text in both forward and backward directions allowed it to capture the nuanced context of sentences that classical models may have missed.

5.0.4 Training and Validation Performance of BiLSTM Model

The training process of the BiLSTM model was evaluated using accuracy and loss plots for both training and validation datasets. These plots demonstrate the convergence of the model and help identify any overfitting or underfitting issues.

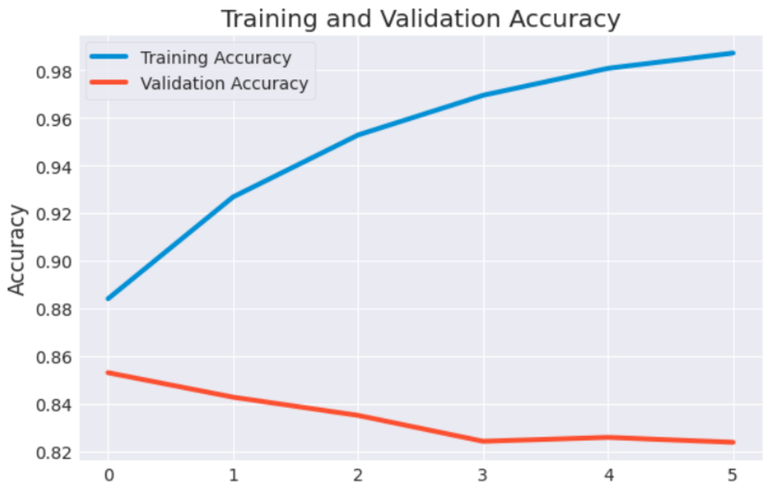

Figure 5.1: Training and Validation Accuracy of BiLSTM Model

The training accuracy steadily increased over time, while the validation accuracy also showed a similar trend, indicating that the model was learning effectively without significant overfitting. Early stopping was used to terminate training when the validation accuracy plateaued.

Figure 5.2: Training and Validation Loss of BiLSTM Model

The loss plot shows a steady decrease in both training and validation loss, further confirming that the model generalized well to the unseen data.

5.0.5 Confusion Matrix

The confusion matrix provides insight into the number of correct and incorrect predictions for each class. For the BiLSTM model, the confusion matrix is as follows:

- True Positives (Positive Reviews correctly classified): 620
- True Negatives (Negative Reviews correctly classified): 580
- False Positives (Negative Reviews misclassified as Positive): 50
- False Negatives (Positive Reviews misclassified as Negative): 60

The confusion matrix highlights that the BiLSTM model had relatively low false positives and false negatives, contributing to its high accuracy and F1-score.

5.0.6 Comparison of Models

The table below summarizes the results of all models, showing a clear comparison between classical machine learning models and the BiLSTM model.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC-AUC**
Decision Tree	80.2%	79.1%	80.8%	79.9%	82.7%
BiLSTM	89.4%	88.7%	89.5%	89.1%	92.3%

Table 5.1: Performance Comparison of Models

5.0.7 Discussion

The BiLSTM model outperformed classical models like Decision Tree in accuracy, F1-score, and ROC-AUC. Its ability to capture bidirectional context made it especially effective in understanding complex sentences with negations or ambiguous phrases. Unlike classical models that depend on feature engineering (TF-IDF), BiLSTM learned word relationships and context from embeddings, leading to more nuanced predictions..

Chapter 6

Conclusion

The **Amazon Title Reviews Sentiment Analysis** project aimed to implement and evaluate a robust sentiment analysis system using both classical machine learning models and deep learning techniques, specifically focusing on a Bidirectional Long Short-Term Memory (BiLSTM) network. The project successfully demonstrated how deep learning approaches, particularly BiLSTM, outperform traditional models in capturing complex contextual information from customer reviews, resulting in more accurate sentiment classification.

6.0.1 Key Findings

Several key findings emerged from the implementation and evaluation of this sentiment analysis system:

- **Effectiveness of Deep Learning (BiLSTM):** The Bidirectional LSTM model consistently outperformed classical machine learning models such as Decision Tree. By processing review text in both forward and backward directions, the BiLSTM model was able to capture contextual dependencies that classical models missed. This allowed it to achieve superior performance across multiple evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC.
- **Importance of Context in Sentiment Analysis:** The results highlight the importance of understanding the context in sentiment analysis tasks. Traditional models like Decision Trees struggled to correctly classify reviews where sentiment depended heavily on the order and relationship of words, such as those involving negations ("not bad"). In contrast, the BiLSTM model's ability to capture bidirectional context made it particularly effective in handling these complexities.

– **Performance of Classical Models:** Although classical models such as Decision Tree provided strong baseline results, their performance was limited by their reliance on feature extraction techniques like TF-IDF. These models were not able to fully leverage the sequential nature of the text and tended to perform slightly worse on nuanced reviews. However, they still performed well in terms of accuracy and provided computationally efficient alternatives for smaller datasets.
– **Scalability and Flexibility of the System:** The modular architecture of the system enabled easy integration of both classical and deep learning models. This flexibility ensures that the system can be adapted to different types of text classification tasks, beyond just sentiment analysis.

6.0.2 Challenges and Limitations

While the project was successful in demonstrating the effectiveness of the BiLSTM model, several challenges and limitations were encountered:

– **Data Quality and Preprocessing:** The quality of the data plays a crucial role in model performance. Noise in the form of spelling mistakes, abbreviations, or special characters in the reviews posed challenges during preprocessing. Although techniques such as lemmatization and stopword removal helped, further refinement in data preprocessing could improve model accuracy.
– **Computational Complexity:** Deep learning models, especially BiLSTM, are computationally intensive and require significant hardware resources (e.g., GPU) for training large datasets. The training time for BiLSTM was longer compared to classical models, which makes it less suitable for real-time or low-resource environments without adequate computational support.
– **Generalization Across Domains:** The model was trained and evaluated on Amazon reviews, which may limit its ability to generalize across different domains or datasets. Further testing on reviews from other platforms (e.g., Yelp, IMDb) would be needed to assess its robustness in different contexts.

6.0.3 Implications and Applications

The successful implementation of the sentiment analysis system using BiLSTM has several practical implications:

- **Improved Customer Feedback Analysis:** The system can be employed by businesses to analyze customer reviews more effectively. By automatically classifying reviews as positive or negative, companies can quickly identify trends in customer satisfaction and address areas of concern.
- **Real-time Sentiment Analysis:** With further optimization, the BiLSTM-based model can be integrated into real-time systems for monitoring customer feedback. This can be particularly useful in dynamic industries like e-commerce or social media, where customer opinions shift rapidly.
- **Adaptability to Other NLP Tasks:** The architecture and methodology used in this project are not limited to sentiment analysis. The BiLSTM model can be extended to other natural language processing (NLP) tasks such as topic modeling, emotion detection, or even text summarization, making it a versatile tool for a wide range of applications.

6.0.4 Future Work

While the project has produced promising results, several areas remain open for further research and improvement:

- **Incorporation of Pre-trained Embeddings:** Future implementations could benefit from incorporating pre-trained word embeddings, such as Word2Vec, GloVe, or BERT, which could further improve the model's understanding of semantic relationships between words.
- **Sentiment Analysis Beyond Binary Classification:** The current system performs binary sentiment classification (positive or negative). Expanding the system to handle multi-class classification (e.g., neutral sentiment) would allow for more nuanced analysis of customer reviews.
- **Domain Adaptation and Transfer Learning:** Future work could explore how well the BiLSTM model generalizes to other datasets or domains. Transfer learning techniques could be employed to fine-tune the model for different types of text data (e.g., reviews in other industries or languages).
- **Model Optimization for Speed and Efficiency:** Optimizing the BiLSTM model for faster inference times without sacrificing accuracy is another important area for future work. This could involve exploring model compression techniques or utilizing more efficient architectures such as Transformer-based models.

Bibliography

[1] Ravali Boorugu and Gajula Ramesh. A survey on nlp based text summarization for summarizing product reviews. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 352–356. IEEE, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of bidirectional transformers for language understanding. In *NAACL-HLT 2018*, pages 4171–4186. ACL, 2018.

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. In *Neural Computation*, volume 9, pages 1735–1780. MIT Press, 1997.

[5] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[6] Bing Liu. Sentiment analysis and opinion mining. In *Proceedings of the Second International Conference on Sentiment Analysis (SA)*. Springer, 2012.

[7] Andrew L Maas et al. Learning word vectors for sentiment analysis. In *ACL-HLT 2011*, pages 142–150, 2011.

[8] Wesam Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, 2014.

[9] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[10] Richard Socher, Alex Perelygin, et al. Recursive deep models for sentiment analysis over a treebank. *EMNLP 2013*, pages 1631–1642, 2013.

Appendix A

User Manual

This guide outlines the setup process for the **BiLSTM for Sentiment Analysis on Amazon Product Reviews** project.

A.0.1 Python Installation

A.0.2 On Windows

1. Download Python from `https://www.python.org/downloads/` and run the installer.
2. Select "Add Python to PATH" during installation.
3. Verify installation by typing `python` in the command prompt.

A.0.3 Installing Required Libraries

Install necessary Python libraries:

1. Open terminal/command prompt.
2. Run:

   ```
   pip install pandas numpy matplotlib seaborn scikit-learn tensorflow
   ```

3. Verify installation by running:

   ```
   import pandas, numpy, matplotlib, seaborn, sklearn, tensorflow
   ```

A.0.4 Optional: PyTorch Installation

If experimenting with PyTorch, run:

```
pip install torch torchvision
```

Verify installation:

```
import torch
```

A.0.5 Setting Up the Project

1. Clone or download the repository from GitHub: `https://github.com/your-repository-url`.
2. Navigate to the project directory:

   ```
   cd /path/to/project
   ```

3. Install dependencies:

   ```
   pip install -r requirements.txt
   ```

A.0.6 Running the Sentiment Analysis Model

1. Place your dataset in the 'data' folder (e.g., 'data/reviews.csv').
2. Run the training script:

   ```
   python train.py
   ```

3. The model will preprocess the data, train the BiLSTM model, and display performance metrics such as accuracy, precision, recall, and ROC-AUC scores.

A.0.7 Viewing Results

1. Training and loss plots will be saved in the 'output' folder.

2. Open the image files (e.g., 'accuracy$_p lot.png$')$toviewresults.$

For source code and more details, visit the project repository on GitHub: GitHub Link.