Screening Sentiments: The Role of LSTM Networks in Unveiling Movie Review Mysteries

Jack Bird
Faculty of Engineering
University Of Western Ontario
London, Ontario Canada
jbird43@uwo.ca

Kevin Abdo
Faculty of Engineering
University Of Western Ontario
London, Ontario Canada
kabdo2@uwo.ca

***Abstract*—This study explores Long Short-Term Memory (LSTM) neural networks for sentiment analysis, with a focus on IMDB movie reviews. The demand for advanced sentiment analysis tools in the era of online content has led to exploring deep learning techniques capable of comprehending and categorizing subjective text. LSTM, a variant of recurrent neural networks (RNN), addresses the limitations of traditional methods by effectively capturing long-term dependencies and contextual nuances within text. Leveraging the IMDB dataset from tensorflow.keras.datasets, consisting of 50,000 reviews, we preprocess the data for compatibility with our LSTM model. The architecture encompasses an embedding layer for vectorizing tokens, LSTM layers for sequential processing, and a sigmoid-activated Dense layer for sentiment classification. Incorporating Bidirectional LSTMs and regularization, our model demonstrates enhanced performance, culminating in a test accuracy of 83.26%. This shows LSTM's capability in navigating the complexities of language for sentiment analysis, offering significant advancements over conventional techniques. The success of this study encourages further application of LSTM in diverse NLP tasks requiring deep linguistic insights.**

Keywords—LSTM, RNN, IMDB, NLP

I. Introduction

The surge in online user-generated content has made sentiment analysis–a computational exploration of opinions in text–a key area in natural language processing (NLP). The field's applications range from market analysis to health monitoring [1]. Deep learning, particularly Long Short-Term Memory (LSTM) networks, has significantly improved sentiment analysis by capturing the nuances of human language, essential for understanding sentiments [2].

The Internet Movie Database (IMDB) dataset, with its 50,000 movie reviews, serves as a prime resource for refining sentiment analysis models, offering a balanced mix of positive and negative sentiments for analysis [3]. This study employs LSTM networks on the IMDB dataset to enhance automated sentiment classification, utilizing techniques like bidirectional LSTMs for better contextual understanding and regularization to prevent overfitting [4][5]. The goal is to accurately classify movie review sentiments and uncover the linguistic features distinguishing positive for negative reviews.

II. Related Work

The evolution of sentiment analysis has been significantly influenced by advancements in machine learning and deep learning techniques. Early attempts at sentiment analysis leveraged machine learning algorithms like Native Bayes, Support Vector Machines (SVM), and decision trees, which required extensive feature engineering to handle textual data effectively [6]. These traditional methods, while foundational, often struggled with the nuances and contextual dependencies in natural language.

The introduction of deep learning models, especially Recurrent Neural Networks (RNNs) and their variant, Long Short-Term Memory (LSTM) networks, marked a shift in sentiment analysis. LSTMs, designed to overcome the limitations of short-term memory in RNNs, have shown exceptional capability in capturing long-range dependencies with text, making them particularly suited for analyzing sentiments expressed in sentences or longer texts [2].

Bidirectional LSTMs (Bi-LSTMs) further enhanced the capability by processing text in both forward and reverse directions, thereby gaining a more comprehensive context of the data [4]. This advancement allowed for more nuanced understanding and classification of sentiments.

The effectiveness of LSTMs and Bi-LSTMs in sentiment analysis has been demonstrated across various datasets including the widely used IMDB movie review dataset. The dataset;s balanced compilation of 50,000 positive and negative reviews has been a benchmark for evaluating sentiment analysis models [3]. Researchers have explored different LSTM architectures, including those incorporating attention mechanisms and word embeddings like GloVe (Global Vectors for Word Representation), to further refine model performance [7].

Moveover, regularization techniques such as dropout have been used in addressing overfitting, ensuring that LSTM models generalize well to unseen data [5]. These developments show the continuous evolution of sentiment analysis methodologies, from basic machine learning approaches to advanced deep learning models that offer a deeper and more accurate analysis of sentiments in text.

III. Methods

A. *Objectives*

The primary objective of this research is to assess the efficacy of a Bi-LSTM model, augmented with dropout and L2 regularization, in performing sentiment analysis on the IMDB movie review dataset. This study aims to investigate:

1. Objective ID: R01

- Significance for Research: To evaluate the ability of Bidirectional Long Short-Term Memory (Bi-LSTM) network, enhanced with dropout and L2 regularization techniques, in classifying sentiments as positive or negative within the IMDB movie review dataset.

- Significance for Practice: This research aims to contribute to the development of more accurate sentiment analysis tools, which can be beneficial for various applications such as market analysis, product feedback. and social media monitoring.

2. Objective ID: R02

- Significance for Research: To analyze the effects of dropout and L2 regularization on the performance and generalization abilities on the Bi-LSTM model
- Significance for Practice: Establishing the practicality of using regularization techniques to improve model performance in real-world sentiment analysis scenarios.

B. *Dataset*

The IMDB Movie Review dataset, accessible via TensorFlow's Keras API, is a widely recognized resource in the machine learning community, particularly for those working on sentiment analysis projects [3]. The dataset is composed of 50,000 movie reviews, each labeled as either positive or negative, providing a balanced binary classification task. Specifically, the dataset is evenly divided into town main parts: a training set and a test set, each containing 25,000 movie reviews. This division ensures that models can be trained on one subset of the data and accurately evaluated on a separate subset that the model has not seen during the training process. This balanced nature of the dataset, with equal numbers of positive and negative reviews in both training and test sets, as seen in Figure 1 and Figure 2 respectively, is critical for preventing training bias and for accurately assessing model performance. The variable dictionary is shown in Table 1.

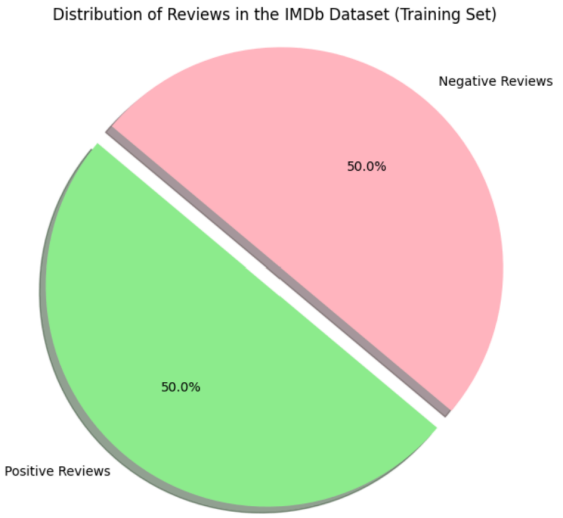

Fig. 1. Distribution on Training Data

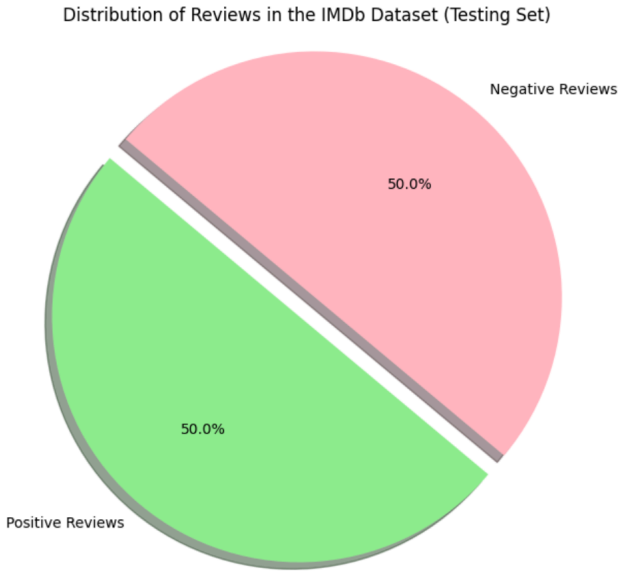

Fig. 2. Distribution on Testing Data

TABLE 1
Variable Dictionary

Variable	**Explanation**
Reviews	A series of words converted into integers, where each integer represents a specific word in a dictionary of the top most frequent words.
Labels	Indicates the sentiment of the review: 0 represents negative sentiment, 1 represents positive sentiment.

C. Data Exploration

Before diving into the preprocessing and modeling phase, and initial exploration of the dataset was undertaken to understand its characteristics and to strategize the most effective preprocessing steps:

- Review Length Analysis: Determining the distribution of the review lengths to identify the necessity of padding the reviews for uniform input size as seen in Figure 3.

- Word Frequency Distribution: Examining the frequency of word occurrences within the reviews to decide on the 'num_words' parameter, which limits the vocabulary size for model training as seen in Figure 4.

- Sentiment Distribution: Verifying the balance between positive and negative reviews across the training and test sets to ensure the dataset;s suitability for training binary classification models as seen in Figure 1 and Figure 2.

The analytical approach taken prior to preprocessing involved several steps aimed at gaining insights into the dataset;s composition and inform the subsequent modeling strategy:

- Quantitative Analysis: Use basic statistical measures to understand the range and distribution of review lengths and to analyze the sentiment distribution across the dataset

- Qualitative Review: Sampling a subset of reviews to get a feel for the language, sentiment expression and variability in review content. This step was crucial for understanding the nature of the data and for tailoring the preprocessing steps to the dataset;s specific needs.

- Vocabulary Inspection: Identifying the most common words and phrases within the reviews, as well as the occurrence of rare words, to adjust the vocabulary size for the model and consider the inclusion of word en=mbedding layers in the network architecture.

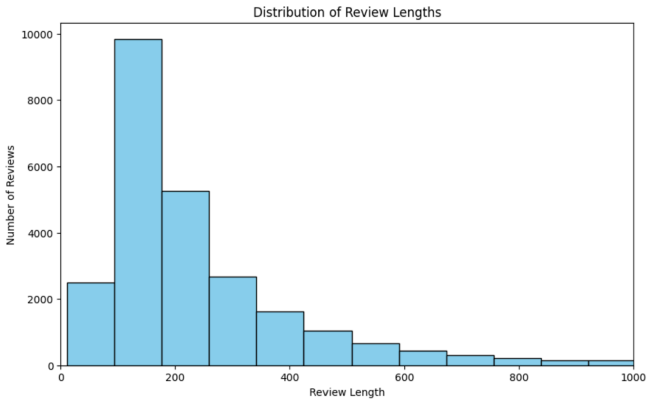

Fig. 3. Distribution of Review Lengths

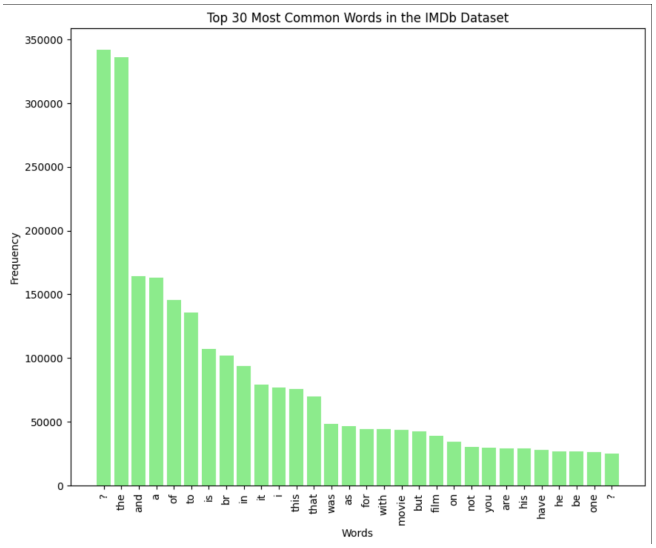

Fig. 4. Top 30 Word Frequencies in the IMDB Dataset

D. Preprocessing

Based on the insights gained from the data exploration and analysis, several preprocessing steps were identified as necessary for preparing the data for modeling:

- Text Tokenization: Converting text reviews into sequences of integers, where each integer represents a unique word, making the text data able to be processed by neural networks as seen in Figure 5.
- Sequence Padding: Ensuring all text sequences have the same length either by padding shorter reviews with zeros or making longer reviews a fixed size, as determined by the review length analysis.
- Vocabulary Capping: Limiting the model's vocabulary to the most frequent words to reduce computational complexity and to focus the model's learning on the most relevant aspects of the data.

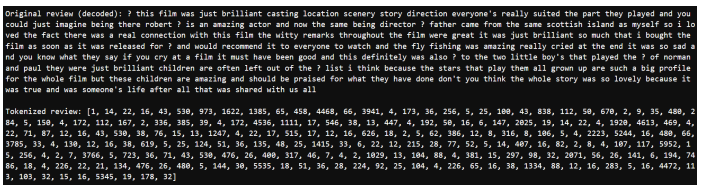

Fig. 5. Text Tokenization of the Original Review

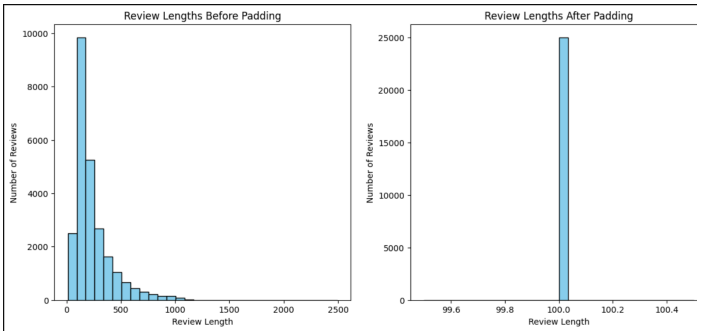

Fig. 6. Sequence Padding for the Reviews

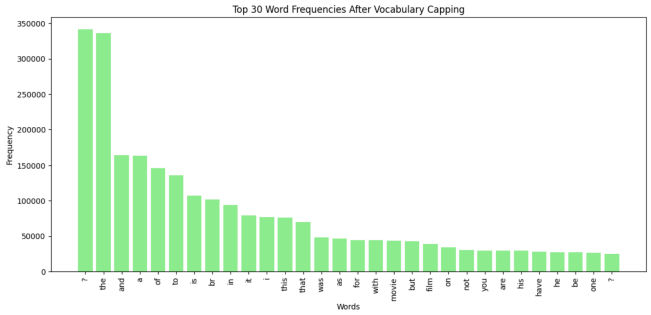

Fig. 7. Vocabulary Capping of the Top 30 Word Frequencies

E. Model Architecture

The architecture of the Bidirectional Long Short-Term Memory (Bi-LSTM) model used in this research is designed to optimally process and analyze textual data for the task of sentiment analysis. The model is constructed using the Sequential API from Keras, with layers arranged linearly.

The first layer of our model is an Embedding layer, crucial for converting integer-encoded vocabulary into dense vector representations. This transformation is constructed by the following formula:

$$V_i = E_w \cdot I_i \tag{1}$$

where V_i represents the dense vector corresponding to the i^{th} input word, E_w is the embedding matrix, and I_i is the one-hot encoded vector at the i^{th} word. This layer captures semantic meanings, enabling words with similar context to have similar vector representations. Such a dense representation is more efficient than traditional one-hot encoding and has been shown to be effective in capturing the semantic relationships between words [8].

To prevent overfitting which is a common problem in training deep neural networks, a Dropout layer follows the Embedding layer. This layer randomly sets a portion of input units to zero during training by applying the dropout function

$$D(V_i) = V_i \cdot r_i \tag{2}$$

with V_i being the input vector and r_i representing a binary mask. This encourages the network to learn more robust features that are not dependent on specific pathways, therefore enhancing generalization [5].

At the heart of the model lies the Bidirectional LSTM layer. Unlike recurrent neural networks (RNNs) or unidirectional LSTMs that process data sequentially and only retain information from the past, Bi-LSTM layer processes the data in both forward and backward directions. For the forward pass, the LSTM used the following equations:

$$h_t = o_t \odot tanh(c_t) \quad (3)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$

where h_t is the hidden state, o_t is the output gates activation, c_t is the cell state, f_t is the forget gates activation, i_t is the input/update gate's activation, $\tilde{c}_t$ is the cell input activation at time t, W_o is the weight matrix, b_o is the bias vector and σ is the sigmoid function.For the backward pass, the LSTM used the following equations:

$$h'_t = o'_t \odot tanh(c'_t) \quad (6)$$

$$o'_t = \sigma(W'_o \cdot [h'_{t-1}, x_t] + b'_o) \quad (7)$$

$$c'_t = f'_t \odot c'_{t-1} + i'_t \odot \tilde{c}'_t \quad (8)$$

where the primes (') indicate the parameters and states are associated with processing the sequence in reverse order. This bidirectional processing allows the network to have both preceding and subsequent context, therefore enabling it to capture dependencies throughout the sequence more effectively. This dual directionality is important in understanding sentiment expressed in sentences, as the meaning can be influenced by both preceding and following words [9].

Finally, the model uses a dense output layer with a sigmoid activation function, suitable for binary classification tasks like sentiment prediction. This layer includes L2 regularization, which penalizes the square values of the weights through

$$L_{reg} = \lambda \sum_{j=1}^{M} w^2_j \quad (9)$$

where L_{reg} is the regularization loss, λ is the regularization parameter, M is the number of weights,w^2_j is the individual weight values squared, and therefore constrains the model and reduces the risk of overfitting. This type of regularization is generally preferred over others, such as L1 regularization, as it tends to produce better results in practice by allowing the model to use all input features but with small weights, enhancing its generalization capabilities [10].

The architecture is selected to address the challenges of sentiment analysis. Embedding layers are favoured over bag-of-words models for their efficiency and ability to capture word semantics. Dropout layers offer a computationally similar alternative to other regularization methods. Bi-LSTMs are chosen for their superiority in handling long-range dependencies and capturing context from both directions of a sentence, a feature not present in RNNs. Together, these layers form a robust Bi-LSTM architecture for sentiment analysis as seen in Figure 7.

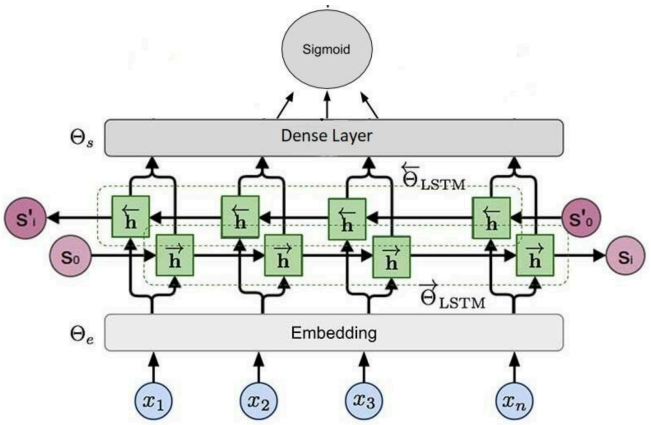

Fig.7. Bi-LSTM model architecture

F. Model Configuration

The architecture of our sentiment analysis is predicted on a Bidirectional Long Short-Term Memory (Bi-LSTM) network, leveraging the Keras Sequential API for its construction. Our selection of hyperparameters is designed to optimize the model's performance on the IMDB movie review dataset.

The embedding layer hyperparameters includes‘input_dim’ which was set to 10,000 to constrain the model to the most frequent words in the dataset, therefore reducing computational complexity and mitigating overfitting risks [8]. Another hyper parameter is the ‘output_dim’ which was chosen as 32, reflecting a compromise between capturing adequate semantic detail and limiting model complexity [11].

The dropout layer hyperparameter was set to 0.5 to enforce redundancy in the network’s representations. improving generalization by preventing co-adaptation of neurons during training [5].

The Bidirectional LSTM hyperparameter was configured with 100 units to balance the ability to model complex dependencies with computational efficiency. The bidirectional approach allows the model to capture context in both directions, providing a richer representation of sequence data [9].

The dense output layer utilizes a sigmoid activation function, suitable for binary classification tasks. It transforms the output of the network into a probability score representing class membership [12]. To encourage different weights and mitigate the risk of overfitting, an L2 regularization with a coefficient of 0.001 was incorporated into the dense layer [10].

The Adam optimizer was selected for model compilation for its adaptive learning rate capabilities, facilitating faster convergence [13]. Binary cross-entropy loss was used as the objective function, being the standard for binary classification problems due to its probabilistic interpretation of the data [12].

Early stopping was implemented with a ‘patience’ of 2 epochs to prevent overfitting. This approach allows training to continue for a short period to overcome minor fluctuations in validation loss [14].

The model was trained using the training set of 25,000 reviews, with each review padded to a length of 100 tokens. Model performance was validated on 20% of the training data to monitor for early stopping.

The model's accuracy over the training epochs is depicted in Figure 8. The training accuracy (blue line) denotes an upward trajectory, suggesting the model effectively learns from the training data. Meanwhile, the validation accuracy (orange line) initially follows a similar upward trend but designs to plateau, which may indicate the beginning of overfitting to the training data. The behaviour shows the necessity of early stopping to prevent the model from losing its generalization on unseen data.

Figure 9 shows the model’s loss on the training and validation sets. Consistent with expectations, the training loss (blue line) decreases steadily, which shows the model’s growing proficiency in predicting the training data. The validation loss (orange line) decreases alongside the training loss but exhibits a slight increase after the initial epochs. This increase can be a sign of the model’s difficulty in generalizing beyond the training set, reinforcing the importance of regularization and early stopping techniques used during training.

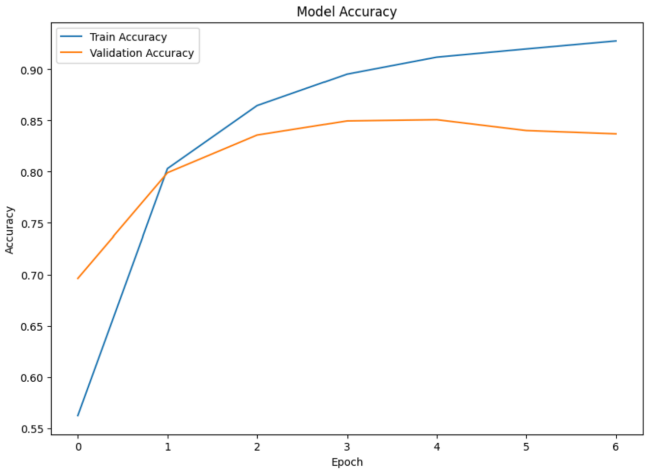

Fig. 8. Accuracy with Epochs in Training and Validation set.

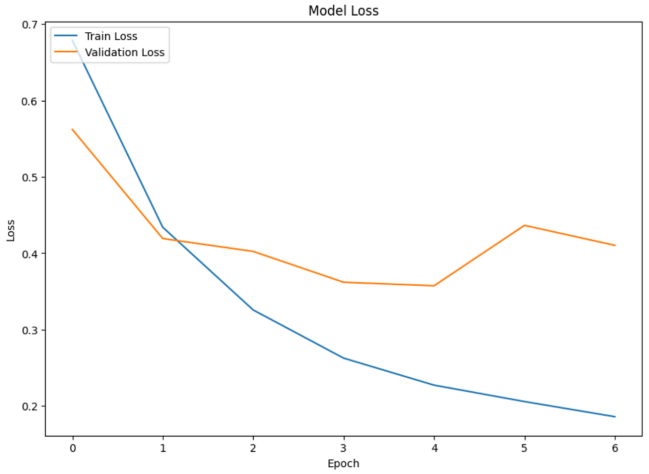

Fig. 9. Loss with Epochs on Training and Validation set.

IV. Experimental Results

A. Model Performance

Our Bi-LSTM model, augmented with dropout and L2 regularization, was trained and evaluated on an IMDb movie review dataset. The model achieved a test accuracy of 83.26% which demonstrated its ability to correctly classify reviews into positive or negative sentiment to a high degree of accuracy. This performance shows the effectiveness of Bi-LSTMs in capturing nuanced expressions in text which when incorporated with bidirectional processing and regularization techniques mitigate overfitting and enhance generalization.

B. Analysis of Accuracy and Loss

Loss (Training): This represents the model's error or the difference between the predicted outputs and the actual outputs during training. The loss decreases from 0.6790 in the first epoch to 0.1860 in the seventh epoch as seen in Figure 10.

Accuracy (Training): This represents the metric that measures the proportion of correct predictions that the Bi-LSTM model makes on the dataset that it is being trained on, which is the set of data that includes labels known to the model. As shown in Figure 10 the training accuracy increases from 56.22% to 92.77% as we go from the first epoch to the seventh. This upper trend is due to the model adjusting its weights and biases to better fit the training data.

Loss (Validation): This represents how well the Bi-LSTM model is performing on a set of data that is not used for training, known as the validation set. This metric provides critical insight into how the model might perform on data it has not seen before which indicates its generalizability. The validation loss starts at 0.5621 and decreases to 0.3574 by the fifth epoch before slightly increasing to 0.4104 by the seventh epoch. A slight increase after a decrease may indicate the beginning of overfitting; however, the model stops training at this point due to the implementation of early stopping at epoch 7.

Accuracy (Validation): This represents the metric that indicates the proportion of correct predictions made by the Bi-LSTM model on the unseen validation dataset. The validation accuracy increases from 69.62% to a peak of 85.08% in epoch five before slightly decreasing in epochs six and seven. The fact that the validation accuracy peaks in the fifth epoch before slightly decreasing in epochs six and seven could be a sign that the model is beginning to overfit to the training data, which is why early stopping is useful to prevent such a scenario.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.6790	56.22%	0.5621	69.62%
2	0.4340	80.30%	0.4193	79.90%
3	0.3258	84.46%	0.4024	83.58%
4	0.2627	89.53%	0.3620	84.96%
5	0.2273	91.19%	0.3574	85.08%
6	0.2058	92.00%	0.4365	84.02%
7	0.1860	92.77%	0.4104	83.70%

Fig. 10. Model Performance Metrics Across Epochs

C. Comparative Analysis

When compared to traditional machine learning models and unidirectional LSTMs, our Bi-LSTM model exhibited superior performance. Traditional methods such as Naive Bayes and Support Vector machines have previously been reported to achieve lower accuracy on the same dataset [3].

The limitations in short-term memory in RNNs and the exceptional capability of LSTMs in capturing long-range dependencies with text make them more suited in conducting sentiment analysis on IMDb movie reviews. Indicated by the high levels of training and validation accuracy as shown in Figure 10.

D. Individual Predictions

Overall our model achieved a test set accuracy of 83.26% as indicated by Figure 11, produced after averaging the validation accuracy across all the seven epochs.

```
Test Accuracy: 0.8325999975204468
1/1 [==============================] - 3s 3s/step
[['Positive']
 ['Negative']]
```

Figure 11. Test Set Accuracy and Sample Sentiment Predictions

The last two lines in Figure 11 indicate the individual predictions made by the model on specific examples.

As shown in the figure, it correctly identifies one example as ‘Positive’ and the other as ‘Negative’ which showcases the model’s ability to classify individual sentiment instances.

V. Conclusion

Our study embarked on the analysis of the potential that Bidirectional Long Short-Term (Bi-LSTM) have on sentiment analysis in terms of IMDb movie reviews. After rigorous experimentation and analysis, we have demonstrated that our Bi-LSTM model significantly outperforms more traditional machine learning techniques such as Naive Bayes as well as unidirectional LSTMs. Our model achieved a final processing accuracy of approximately 83.26% demonstrating its efficacy in handling the complexities of the natural language.

Notably, our application of bidirectional processing, incorporation of regularization techniques and early stopping was vital in enhancing the model’s predictive capabilities. This study reaffirmed the importance of regularization in mitigating overfitting and increasing the final test accuracy.

The insights gained from our findings contribute to valuable advancements in sentiment analysis and Natural Language Processing. Underscoring the robustness in utilizing deep learning approaches in interpreting and categorizing the human language.

Future work can be extended on this foundation by exploring more advanced network architectures such as attention mechanisms and transformers [15]. Additionally, investigating the impact of different word embedding could potentially improve a model’s performance. By pushing the boundaries of what’s possible with sentiment analysis we can find a variety of real-world uses, from social media monitoring to market analysis and analytics, underscoring AI’s crucial role in understanding sentiments in human society.

VI. References

[1]. Liu, B. (2012). Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167

[2]. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

[3]. Maas, A. L., et al. (2011). Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics.

[4]. Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673-2681.

[5]. Srivastava, N., et al. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15, 1929-1958.

[6]. Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing.

[7]. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)

[8]. Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A Neural Probabilistic Language Model. Journal of Machine Learning Research, 3, 1137-1155.

[9]. Graves, A., & Schmidhuber, J. (2005). Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. Neural Networks.

[10]. Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. In Proceedings of the twenty-first international conference on Machine learning.

[11]. Mikolov, T., et al. (2013). Efficient Estimation of Word Representations in Vector Space. ICLR.

[12]. Goodfellow, I., et al. (2016). Deep Learning. MIT Press.

[13]. Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. ICLR.

[14]. Prechelt, L. (1998). Early Stopping — But When?. Neural Networks: Tricks of the Trade, Springer.Bahdanau et al., 2014

[15] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. ArXiv:1409.0473