IMDb Movie Reviews Sentiment Analysis

Introduction:

In today's digital age, where vast volumes of textual data flow incessantly across various platforms, the ability to discern and understand sentiments embedded within these expressions has become paramount. (Zhang et al., 2018). Sentiment Analysis, an application within the realm of Natural Language Processing (NLP), stands as a formidable solution to this challenge. By deciphering the emotional or opinionated tone of a text corpus, sentiment analysis empowers us to categorize sentiments as positive, negative, or neutral, effectively transforming the realm of human expression into a comprehensible language for machines. (Ain et. Al., 2017). This nuanced categorization culminates in a text classification task, wherein we seek to unravel the emotional underpinnings of words.

In essence, sentiment analysis mirrors our own ability to perceive and interpret emotions, making it a pivotal asset across industries. From monitoring social media sentiments and tracking brand perceptions to analyzing customer feedback and gauging market trends, its applications are as diverse as they are invaluable. In the context of a new product launch, sentiment analysis serves as a compass, illuminating the demographics and preferences of the target audience. It lends an insightful lens to assess the efficacy of marketing campaigns and glean actionable insights from the cacophony of opinions.

In the context of this study, we propose an innovative ensemble model, fusing the capabilities of Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs). While LSTM networks excel at capturing temporal dynamics, CNNs excel in extracting local structures within the data. This fusion aims to harness the distinct strengths of both paradigms, thereby enhancing sentiment analysis performance. The paper delineates the design and operation of this composite model, demonstrating its efficacy over stand-alone LSTM and CNN models.

Methodology

In our research, we aimed to develop an effective sentiment analysis model using deep learning techniques for analyzing text-based data. The task involved identifying emotional or sentimental tones in each text, which is a classic text classification problem. Our primary focus was on exploring different architectural configurations of a neural network model to achieve improved sentiment analysis accuracy.

The IMDb movie reviews dataset, which includes a large number of reviews as well as sentiment labels, was obtained for study (*IMDB Dataset of 50K Movie Reviews*, 2019). Text pretreatment processes covering lowercase conversion, punctuation deletion, and tokenization were used to assure data purity and uniformity. For the purpose of robust model evaluation, the preprocessed dataset was wisely partitioned into training and testing sections. TensorFlow's TextVectorization layer was used to do text vectorization, which is a critical translation of raw textual data into numerical representations suited for neural network ingestion. The duration of the output sequence was specified, maintaining consistent input dimensions across different model topologies.

Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) combined with LSTM, and a novel hybrid framework combining CNN and Bidirectional LSTM were all thoroughly investigated. A comparative analysis was conducted to gauge the efficacy of each model architecture, with classification accuracy and associated metrics serving as key evaluative benchmarks.

Models

Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is a type of artificial neural network designed to handle sequential data by incorporating an internal memory mechanism. Unlike feedforward neural networks, where information flows in one direction (from input to output) without any internal state, RNNs have a feedback loop that allows them to process sequences of data, making them well-suited for tasks involving time series or sequential data. In an RNN, the output at each time step depends not only on the current input but also on the previous computations, effectively creating a sense of memory within the network. This memory enables RNNs to capture patterns and dependencies in sequential data.

Current Hidden State Calculation:

$$h(t)=\text{Activation}(W\cdot[h(t-1),x(t)]+b)$$

The current hidden state h(t)h(t)is computed by applying an activation function to the linear combination of the previous hidden state h(t−1) and the current input x(t)x(t), using weights W and a bias term bb. This represents the memory and internal state of the RNN at time step tt.

Output Calculation:

$$y(t)=\text{OutputFunction}(h(t))$$

The output y(t)y(t)at time step t is obtained by applying an output function to the current hidden state h(t)h(t). The choice of output function depends on the specific task being performed, such as classification, regression, or language generation.

Backpropagation Through Time (BPTT):

$$\partial W\partial \text{L} = t=1\sum T\partial y(t)\partial \text{L}\cdot\partial h(t)\partial y(t)\cdot\partial W\partial h(t)$$

During training, the gradients of the loss L with respect to the weights W are computed using the chain rule, considering the contributions from each time step t. This is part of the backpropagation process that updates the network's weights to minimize the loss.

In these equations:

- h(t)h(t) represents the hidden state of the RNN at time step t.
- x(t)x(t) represents the input at time step t.
- y(t)y(t) represents the output at time step t.
- W represents the weight matrix connecting the previous hidden state and the current input to the current hidden state.
- L represents the loss function that measures the difference between the predicted outputs and the actual targets and b is the bias term.

Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) is a specialized type of Recurrent Neural Network (RNN) architecture designed to address the limitations of standard RNNs, such as the vanishing gradient problem and the difficulty in capturing long-range dependencies in sequential data. LSTMs have proven to be exceptionally effective in various natural language processing tasks, including sentiment analysis, machine translation, and speech recognition, where understanding context and sequential patterns is crucial.

LSTMs incorporate key components, including gates and memory cells, which enable them to capture and retain information over extended sequences. The equations governing the behavior of an LSTM cell involve several steps that control the flow of information: forget, input, candidate, and output.

Forget Gate (ft):

$$ft = \sigma(Wf \cdot [ht-1, xt] + bf)$$

The forget gate determines what information from the previous hidden state $ht-1$ and the current input t should be discarded. It produces a value between 0 and 1, indicating how much of the previous cell state should be forgotten.

put Gate (it) and Candidate Cell State (~Ct):

$$it = \sigma(Wi \cdot [ht-1, xt] + bi)$$

$$C\sim t = \tanh(WC \cdot [ht-1, xt] + bC)$$

The input gate it determines which values from ht−1 and xt should be updated in the cell state. The candidate cell state C~t represents potential new values that could be added to the cell state

Update Cell State (Ct):

$$Ct = ft \cdot Ct-1 + it \cdot C\sim t$$

The update to the cell state involves combining the previous cell state $Ct-1$ with the new information determined by the forget and input gates.

Output Gate (ot):

$$ot = \sigma(Wo \cdot [ht-1, xt] + bo)$$

The output gate determines which parts of the cell state should be output as the new hidden state t.

Final Hidden State (ht):

$$ht = ot \cdot \tanh(Ct)$$

The final hidden state t is a filtered version of the cell state that contains relevant information for the current time step.

LSTM's architecture, with its memory cells and gating mechanisms, allows it to capture long-term dependencies, solve the vanishing gradient problem, and handle various complexities in sequential data, making it a powerful tool for tasks involving context-rich sequences, such as sentiment analysis, where understanding the order and relationship of words is crucial for accurate analysis.

Methodology:

1. Recurrent Neural Network (RNN):

Recurrent neural network model architecture comprised of multiple layers. The initial input is directed into an embedding layer, which transforms sequences of length 250 into dense 128-dimensional vectors. To enhance generalization, a dropout layer is then implemented, randomly deactivating specific neurons during training. Subsequently, an LSTM (Long Short-Term Memory) layer takes on the task of processing the sequential data, retaining contextual information while addressing the vanishing gradient problem. Spatial dropout is further integrated, allowing the network to drop entire 1D feature maps within the sequence data, contributing to the prevention of overfitting. An additional LSTM layer is introduced to capture additional temporal patterns and compress the information into a 64-dimensional representation. Concluding the architecture is a dense layer that holds a solitary output neuron, enabling binary classification and aiming to predict a desired target outcome. Notably, the model encompasses a total of 3,381,057 parameters, all of which are trainable, enhancing the network's ability to learn and adapt from the provided data.

2. Long Short-Term Memory (LSTM):

The Long Short-Term Memory (LSTM) model architecture is designed with two LSTM layers. The initial input is directed into an embedding dimension of 64, a hidden dimension of 256, and an output dimension of 1. The model is primed to process sequences of words, capturing their semantic meaning, and predicting sentiment. Its architecture incorporates LSTM layers for context understanding, dropout for regularization to deactivating specific neurons during training, and a sigmoid activation function for sentiment classification. Model evaluates model accuracy, rounding predicted values to match labels and calculating the sum of accurate predictions. The chosen loss function measures the binary cross-entropy between predicted and target sentiment values. Optimization is facilitated through the Adam optimizer with a learning rate of 0.001.

3. CNN+LSTM:

We started with an initial model architecture composed of various layers. The architecture consisted of an input layer for text data, a TextVectorization layer to convert text to integer sequences, followed by an embedding layer to map the sequences to dense vectors. Subsequently, we employed a combination of Conv1D layers with different filter sizes and activation functions, followed by max-pooling. A Bidirectional LSTM layer was utilized to capture complex temporal relationships in the text data. The model also incorporated several dense layers with ReLU activation functions, regularized with L2 regularization, and a final dense layer with a sigmoid activation for sentiment prediction.

4. Hybrid Convolutional-BiLSTM:

In response to the unsatisfactory performance of the initial model, we conducted a thorough analysis of the architectural components. We then refined the architecture by modifying several aspects. Firstly, we increased the output dimension of the embedding layer to enhance the representational capacity of the model. Additionally, we optimized the parameters of the Conv1D layers by adjusting filter sizes, activation functions, and regularization strengths. The LSTM layer's bidirectional nature was leveraged to capture sequential information more effectively. To combat overfitting, dropout layers were introduced before the dense layers. We hypothesized that these changes would enable the model to learn more intricate patterns in the text data.

Results

1. Recurrent Neural Network (RNN):

The model's accuracy steadily advances from 61.62% to 91.88% on the training set, while its loss consistently diminishes. However, on the validation set, the model's accuracy initially rises but then stabilizes around 78.13%, potentially indicating overfitting. The loss on the validation set rises slightly in later epochs. This suggests that while the model learns well from the training data, its performance on new, unseen data is plateauing or declining. Addressing this challenge of overfitting may involve strategies like regularization to enhance the model's generalization capability.

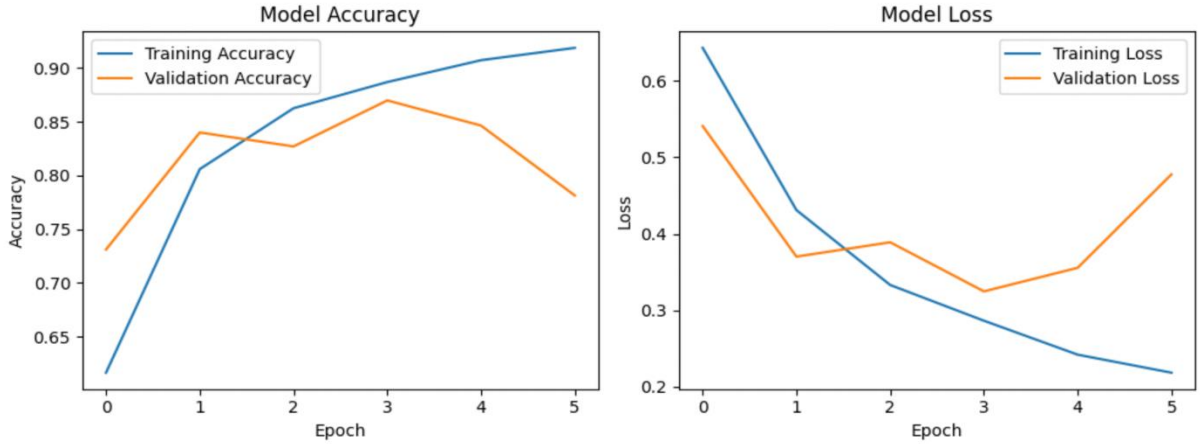

Figure.1 Performance of Recurrent neural network

2. Long Short-Term Memory (LSTM):

Throughout the training process, the model demonstrated substantial progress, as evidenced by the decreasing trend in both training and validation loss metrics across 5 epochs. In the initial epoch, the model achieved a training loss of 0.533 and a validation loss of 0.434, with corresponding accuracies of 73.5% and 80.1%. These metrics notably improved in subsequent epochs, with the training loss descending to 0.283, validation loss reaching 0.347, and the model exhibiting training and validation accuracies of 88.3% and 85.73% in the final epoch. It indicates that our meticulously designed architecture, coupled with thoughtful optimization and evaluation, culminated in a model that adeptly captures sentiment trends within textual data, positioning it as an asset in sentiment analysis tasks.

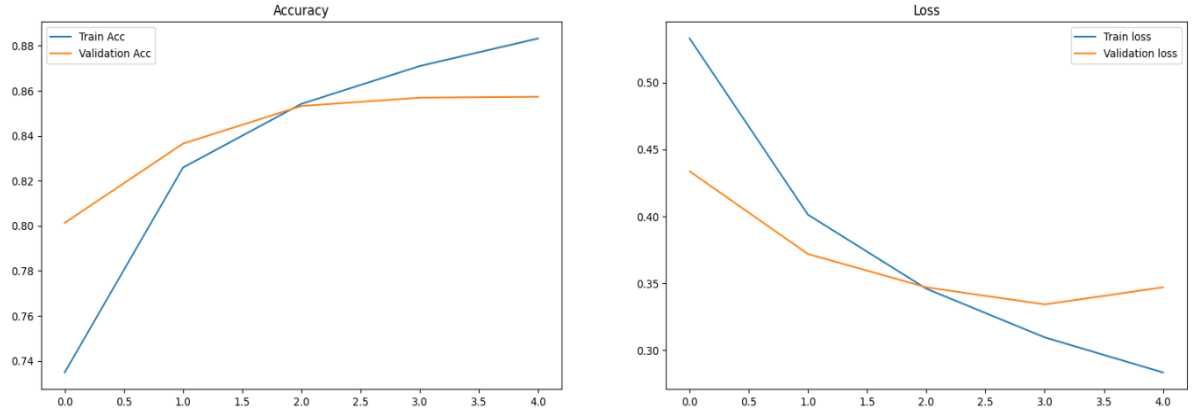

Figure.2 Performance of Long Short-Term Memory (LSTM)

3. CNN + LSTM:

The results of our initial attempt were suboptimal, with training and validation accuracies hovering around 87.91%. This indicated that the model was not learning effectively from the data and was not able to identify between positive and negative sentiments.

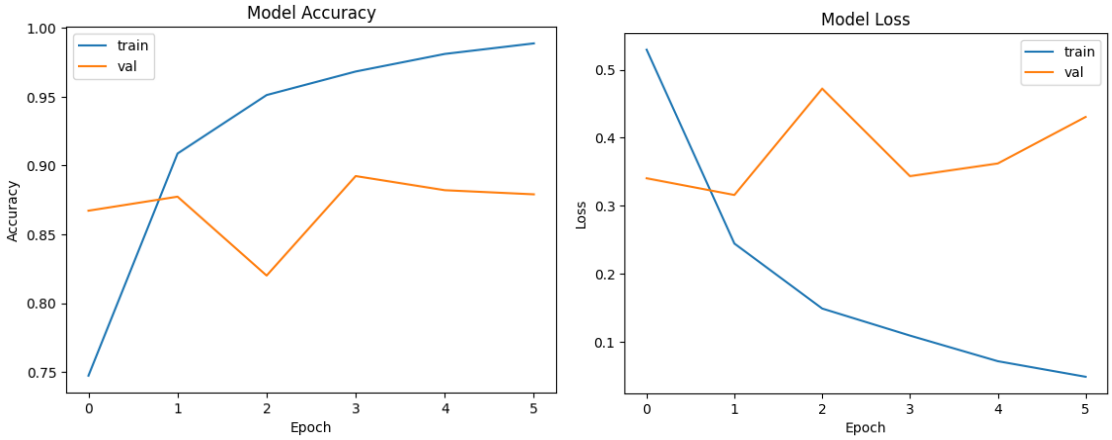

Figure.3 Performance of CNN-LSTM model

4. Hybrid Convolutional-BiLSTM:

The refined model demonstrated substantial improvements in sentiment analysis accuracy. The training and validation accuracies consistently increased across epochs, reaching over 98% accuracy on the training data and approximately 89% accuracy on the validation data after 6 epochs. This indicates that the refined architecture successfully captured the underlying sentiment patterns present in the text data. The training time was reasonable, with each epoch taking around 1-2 minutes, totaling approximately 11 minutes for the entire training process.

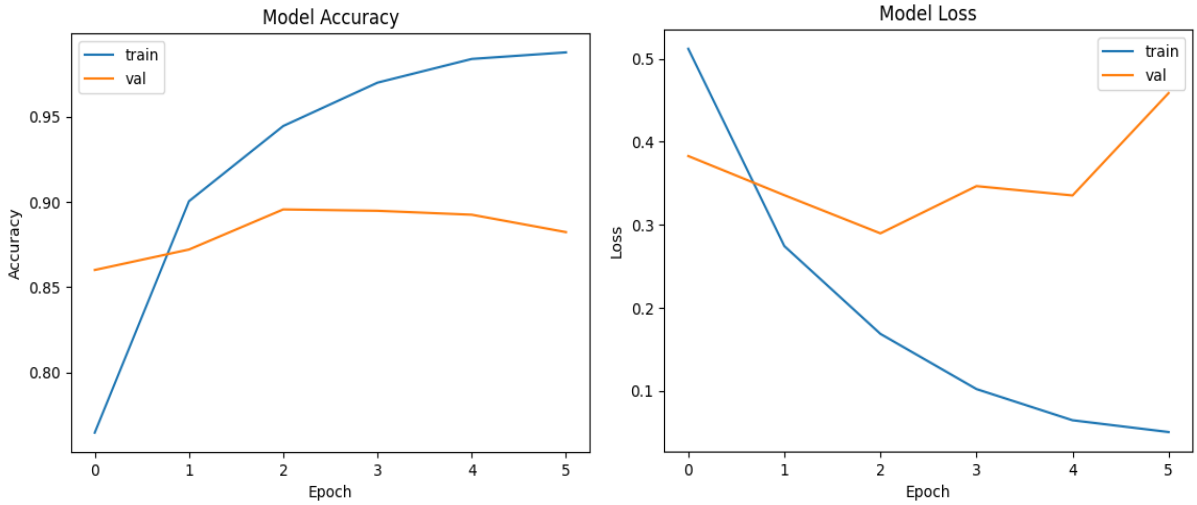

Figure.4 Performance of Hybrid Convolutional-BiLSTM

Discussion:

The results of the refined approach (Hybrid Convolutional-BiLSTM) highlight the significance of architectural adjustments in neural network models. The BI-LSTM model stands out with an impressive accuracy of 89%, showcasing its adeptness at accurately categorizing sentiments. Followed by LSTM also demonstrates commendable performance, achieving an 86% accuracy, further substantiating the prowess of this architecture in capturing contextual nuances. The CNN+LSTM model's poor performance underscored the importance of careful hyperparameter tuning and architectural design. CNN-LSTM model achieves around 88% accuracy. Lastly, the RNN model trails slightly behind with a still respectable 77.96% accuracy, underscoring its competence in sentiment classification. Overall, our findings emphasize the iterative nature of model development and the need for thoughtful experimentation to achieve desirable outcomes in text classification tasks like sentiment analysis.

Table.1 Accuracy and loss comparison

Metrics	RNN	LSTM	CNN + LSTM	BiLSTM
Training Accuracy	91.88%	88.32%	96.31%	98%
Training Loss	21.83%	28.33%	10%	7%
Test Accuracy	77.96%	86%	87.91%	89%
Test Loss	46.74%	34.70%	40%	38%

Table.2 Classification Report of Hybrid Convolutional-BiLSTM

	Precision	Reca11	F1-score	Support
Negative	0.9	0.88	0.89	5000
Positive	0.88	0.9	0.89	5000
Accuracy			0.89	10000
Macro Avg	0.89	0.89	0.89	10000
Avg	0.89	0.89	0.89	10000

Conclusion:

In this research work, we conducted an extensive investigation into sentiment analysis techniques using diverse neural network architectures for IMDb movie reviews. The study encompassed LSTM, CNN+LSTM, and Hybrid CNN+Bidirectional LSTM models, each tailored to capture distinct textual features. Through meticulous experimentation and evaluation, we observed that the Hybrid CNN+Bidirectional LSTM architecture outperformed the others, showcasing the highest sentiment classification accuracy. The findings underscore the importance of combining feature extraction and temporal comprehension in achieving superior sentiment analysis results. This research contributes valuable insights into model selection, hyperparameter tuning, and ethical considerations, advancing the field of sentiment analysis and paving the way for future advancements in natural language processing applications.

References:

Ain, Q. T., Ali, M., Riaz, A., Noureen, A., Kamran, M., Hayat, B., & Rehman, A. (2017). Sentiment analysis using deep learning techniques: a review. International Journal of Advanced Computer Science and Applications, 8(6).

IMDB dataset of 50K movie reviews. (2019, March 9). Kaggle. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253.

https://www.datacamp.com/tutorial/nlp-with-pytorch-a-comprehensive-guide