Sentiment Analysis on the IMDB Dataset using Word2Vec Embedding and Deep Learning Techniques

Shubham Soni[1] and Dr. Natesha B V[2]

Department of Computer Science and Engineering, Indian Institute of Information Technology Raichur, Raichur, India.

Contributing authors: shubham134soni@gmail.com; nateshbv@iiitr.ac.in;

Abstract

Sentiment analysis is a key task in Natural Language Processing (NLP), focused on identifying sentiments in textual data. This project performs sentiment classification on the IMDB movie reviews dataset, which contains 50,000 balanced positive and negative reviews. The objective is to evaluate the performance of deep learning models using semantic word representations. The raw texts were preprocessed through HTML tag removal, lowercasing, special character and stopword elimination, and lemmatization. The cleaned data was converted into 300-dimensional vectors using the pre-trained Word2Vec model from Google News. Reviews were padded to a maximum length of 300 tokens. We trained several deep learning models: CNN, LSTM, two-layer LSTM, Bi-LSTM, and a CNN-LSTM hybrid. Performance was assessed using accuracy, ROC curves, and confusion matrices. CNN achieved the highest accuracy (89.02%), outperforming LSTM (88.05%), CNN-LSTM (88.30%), and Bi-LSTM (87.69%), indicating its effectiveness in extracting local features from text using Word2Vec embeddings.

Keywords: Sentiment Analysis, IMDB Dataset, Natural Language Processing, Deep Learning, Convolutional Neural Networks, Long Short-Term Memory, Bidirectional LSTM, Word2Vec Embeddings, Text Classification, Neural Networks, Machine Learning, Feature Extraction

1 Introduction

With the explosive growth of digital content on the internet, analyzing and understanding textual data has become increasingly important. One such critical task is sentiment analysis, a subfield of Natural Language Processing (NLP) that involves determining the emotional tone or opinion expressed in a piece of text. This technique is widely used in domains such as social media monitoring, customer feedback analysis, product reviews, and opinion mining. In recent years, deep learning methods have shown remarkable success in solving NLP problems by learning complex patterns and contextual relationships from data. In this project, we aim to leverage deep learning techniques for sentiment classification of movie reviews using the IMDB dataset, a well-known benchmark in the NLP community. The dataset contains 50,000 pre-labeled reviews, equally divided between positive and negative sentiments, making it ideal for binary classification tasks.

To accurately capture the semantic meaning of words and sentences, we used pre-trained Word2Vec embeddings from Google News, which map words into a 300-dimensional vector space based on their contextual usage. This approach allows our models to understand the meaning and relationships between words more effectively than traditional Bag-of-Words or TF-IDF techniques. Our results show that CNN outperformed other models, achieving the highest classification accuracy. This indicates that CNN's ability to capture local patterns, when combined with semantic embeddings from Word2Vec, is highly effective for sentiment analysis.

Through this project, we demonstrate the power of combining word embeddings with deep learning architectures for extracting meaningful insights from textual data. The findings provide valuable direction for future work in sentiment analysis and NLP applications using neural networks.

2 Related Work

Sentiment analysis has been widely explored using both traditional machine learning and deep learning techniques. Early studies used models like Logistic Regression, SVM, and Decision Trees, typically relying on text representations such as TF-IDF and Bag-of-Words. While effective for simple tasks, these models lacked the ability to understand semantic relationships in text.

With the introduction of word embeddings like Word2Vec, deep learning models such as CNN, LSTM, and Bi-LSTM became popular. CNNs are effective in extracting local patterns, while LSTMs capture long-term dependencies in text sequences. Hybrid models like CNN-LSTM and Bi-LSTM further improved performance by combining spatial and sequential feature learning.

More recent research has focused on transformer-based models like BERT and RoBERTa, which provide contextual embeddings and have shown strong performance in sentiment analysis tasks. These models leverage self-attention mechanisms to understand the deeper meaning of text.

Overall, existing work demonstrates a shift from traditional approaches to deep neural networks and transformer-based methods, emphasizing the importance of semantic and contextual understanding in sentiment classification.

3 Dataset and Preprocessing

3.1 IMDB Movie Reviews Dataset

In this project, we use the IMDB movie review dataset from Kaggle, which contains a total of 50,000 reviews, equally divided into 25,000 positive and 25,00 negative samples. This dataset is widely recognized and frequently used in binary sentiment classification tasks within the field of Natural Language Processing (NLP).

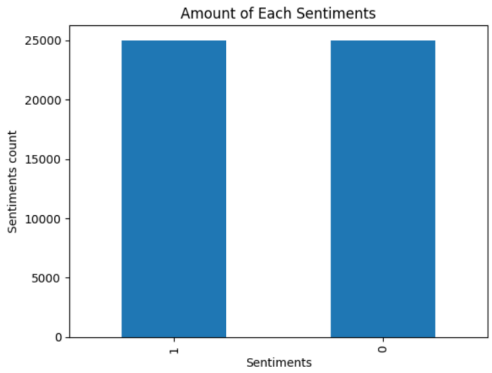

Fig. 1 Sentiment distribution in the IMDB dataset

For our experiments, we divided the dataset into training and testing sets with an 80:20 split, maintaining the balance between positive and negative reviews in both sets. This resulted in 40,000 reviews for training and 10,000 reviews for testing.

3.2 Text Preprocessing

Raw text data typically contains noise, irrelevant information, and formatting inconsistencies that can adversely affect model performance. Therefore, comprehensive preprocessing is a crucial step in NLP tasks. We implemented a sequential preprocessing pipeline to clean and standardize the text data:

- **Fixing Contractions:** Contractions such as ‘‘don’t’’ and ‘‘isn’t’’ are expanded to ‘‘do not’’ and ‘‘is not’’, helping standardize the text and reduce ambiguity.
- **Removing HTML Tags:** HTML tags (e.g.,
, <i>) embedded in the reviews are removed to retain only the relevant text content.
- **Converting Text to Lowercase:** All characters are converted to lowercase to avoid treating the same word differently due to case (e.g., ‘‘Movie’’ and ‘‘movie’’ are made identical).
- **Removing Special Characters and Punctuation:** All non-alphanumeric characters, punctuation marks, and symbols are eliminated to reduce noise and improve model focus on meaningful content.
- **Stopword Removal:** Frequently occurring but semantically weak words like ‘‘the’’, ‘‘is’’, and ‘‘and’’ are removed using the NLTK English stopword list.
- **Lemmatization:** Using WordNet lemmatizer, words are reduced to their root forms (e.g., ‘‘running’’ becomes ‘‘run’’, ‘‘movies’’ becomes ‘‘movie’’), which helps reduce dimensionality and improve semantic analysis.

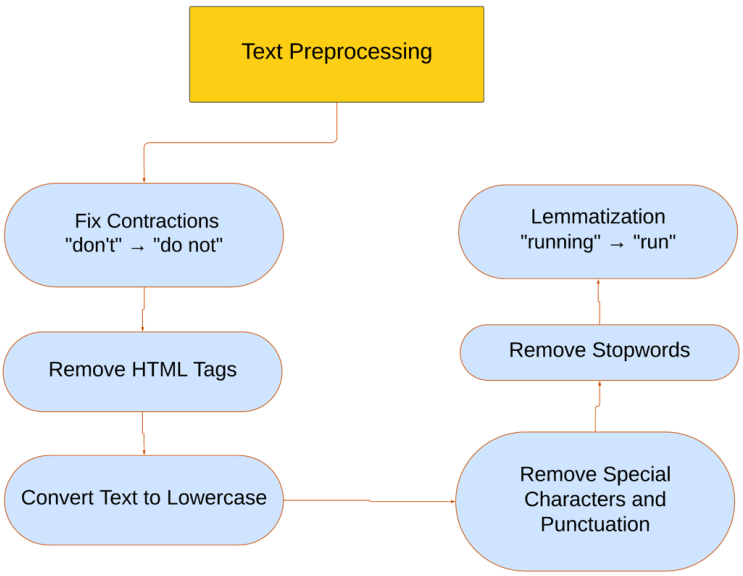

Fig. 2 Steps involved in text preprocessing

These preprocessing techniques collectively transform raw, noisy text into clean and structured inputs suitable for neural network-based sentiment analysis.

3.3 Data Standardization

To ensure consistent input dimensions for our neural network models, we standardized the length of all reviews. After analyzing the distribution of review lengths in the dataset, we established a maximum sequence length of 300 tokens. Reviews longer than this limit were truncated, while shorter reviews were padded with zeros. This standardization process is essential for batch processing in neural networks, as it allows for efficient computation and consistent feature representation.

The choice of 300 tokens as the maximum length was based on a balance between preserving sufficient information and computational efficiency. Our analysis showed that this length captured the majority of content in most reviews while keeping the input dimensions manageable for model training.

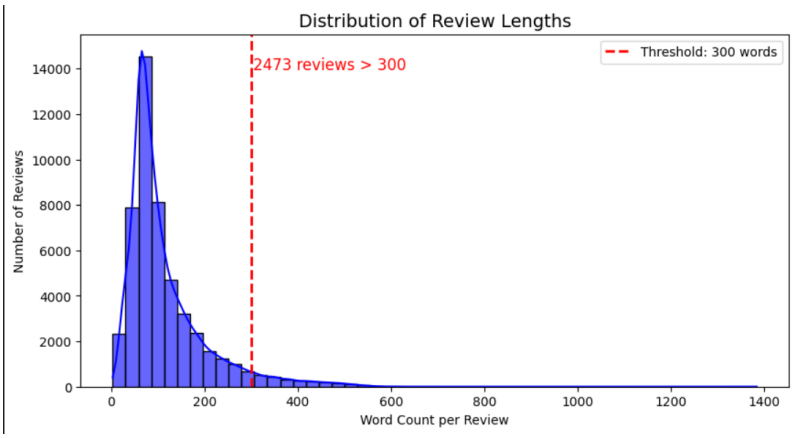

Fig. 3 Distribution of review lengths

4 Word2Vec Embedding

In this project, we used the pre-trained Word2Vec model word2vec-google-news-300 provided by the Gensim API. This model was trained on approximately 100 billion words from the Google News dataset, making it a robust and widely-used choice for generating word embeddings. Each word in the vocabulary is represented as a dense 300-dimensional vector, capturing semantic and syntactic relationships between words. The model includes a vocabulary of about 3 million words and phrases, ensuring broad language coverage.

One of the key properties of Word2Vec embeddings is that words with similar meanings or contextual usage tend to have similar vector representations, enabling more effective feature extraction for natural language processing tasks such as sentiment analysis, text classification, and similarity detection.

Several word embedding techniques have been developed, including Word2Vec, GloVe , and FastText . While we experimented with both TF-IDF vectorization and Word2Vec embeddings in the initial stages of our research, we ultimately chose Word2Vec for its superior ability to capture semantic relationships between words.

4.1 Handling Out-of-Vocabulary Words

One challenge with using pre-trained word embeddings is dealing with out-of-vocabulary (OOV) words—words that are present in our dataset but not in the pre-trained model's vocabulary. These may include movie-specific terminology, names, or neologisms. For OOV words, we assigned zero vectors (vectors with all components set to zero), effectively treating them as neutral elements in the vector space.

5 Deep Learning Models

In this section, we detail the various deep learning architectures implemented for sentiment analysis. Each model was designed to capture different aspects of the text data, from local patterns to long-range dependencies. All models shared common hyperparameters to ensure fair comparison: embedding dimension $d = 300$, sequence length $L = 300$, training-testing split of 80:20, 25 epochs for training, and batch size of 64.

5.1 CNN (Convolutional Neural Network)

Convolutional Neural Networks have shown remarkable success in text classification tasks . While originally developed for image processing, CNNs are effective for text analysis due to their ability to detect local patterns and n-gram-like features regardless of their position in the text. The CNN model focuses on capturing local patterns in the text through convolutional filters. It identifies important n-gram features regardless of their position in the sequence. Despite being less sequential than LSTMs, the CNN model achieved the highest test accuracy of 89.02%, indicating its strength in identifying critical features in sentiment analysis tasks. Figures 4 and 5 illustrate the performance of the Bi-LSTM model using the confusion matrix and ROC curve respectively.

5.2 LSTM (Long Short-Term Memory)

Long Short-Term Memory networks are specifically designed to capture sequential patterns and long-range dependencies in text. LSTMs belong to the family of Recurrent Neural Networks (RNNs) but include a sophisticated gating mechanism that allows them to selectively remember or forget information, addressing the vanishing gradient

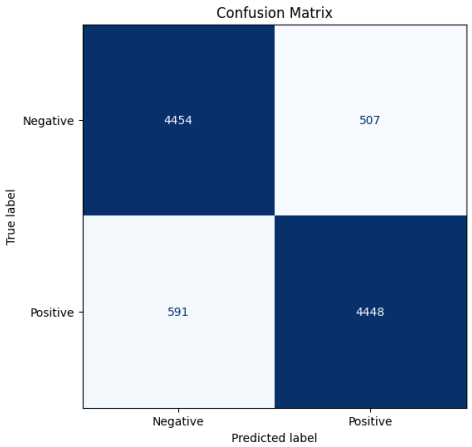

Fig. 4 Confusion matrix for CNN

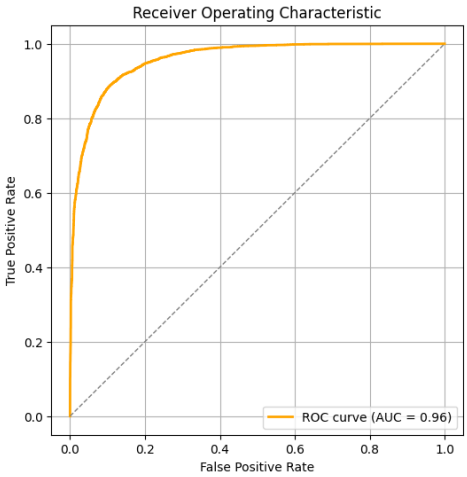

Fig. 5 ROC curve with AUC = 0.96 for CNN

problem that plagued traditional RNNs. The LSTM model is effective at capturing long-range dependencies in sequential data. It processes input tokens sequentially, maintaining a memory of previous words, which makes it suitable for sentiment classification. The LSTM model achieved a test accuracy of 88.05%, demonstrating solid performance on the dataset by learning the temporal structure of reviews. Figures 6 and 7 illustrate the performance of the Bi-LSTM model using the confusion matrix and ROC curve respectively.

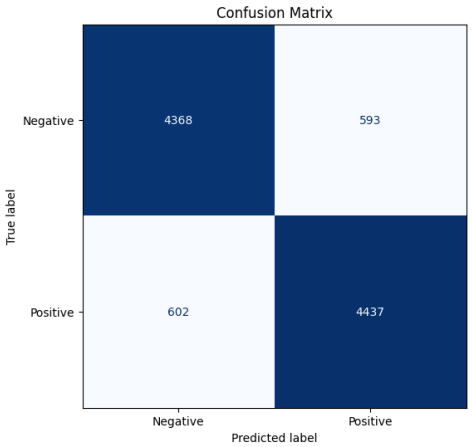

Fig. 6 Confusion matrix (LSTM)

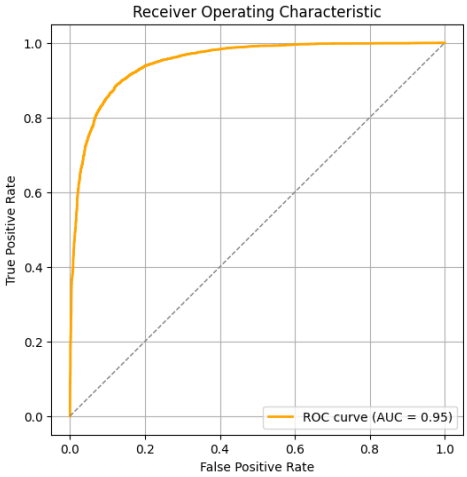

Fig. 7 ROC curve with AUC = 0.95 (LSTM)

5.3 Bi-LSTM (Bidirectional LSTM)

Standard LSTMs process text in a forward direction, from the beginning to the end. However, in many NLP tasks, including sentiment analysis, context from both directions (past and future) can be valuable. Bidirectional LSTMs address this by processing the input sequence in both forward and backward directions. However, in this experiment, the Bi-LSTM model achieved a slightly lower accuracy of 87.63%, possibly due to overfitting or the increased complexity not yielding proportional performance gains. Figures 8 and 9 illustrate the performance of the Bi-LSTM model using the confusion matrix and ROC curve respectively.

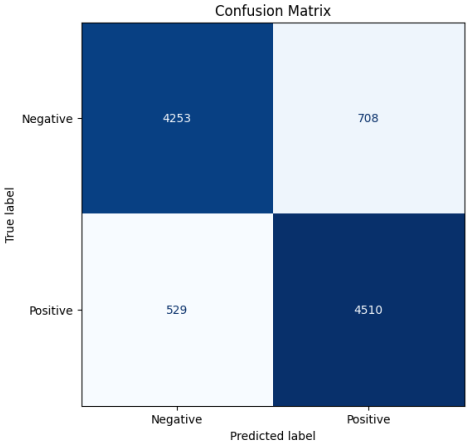

Fig. 8 Confusion matrix (Bi-LSTM)

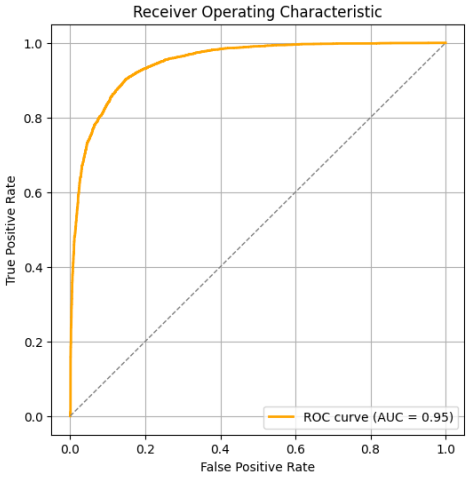

Fig. 9 ROC curve with AUC = 0.95 (Bi-LSTM)

5.4 Bi-LSTM with 2-layer

To enhance representational power, a deeper version of Bi-LSTM with two stacked layers was implemented. This model aimed to extract higher-level sequence features. It showed improved performance over the single-layer Bi-LSTM, achieving an accuracy of 88.52%, indicating that deeper models can capture more nuanced dependencies when properly regularized. Figures 10 and 11 illustrate the performance of the Bi-LSTM model using the confusion matrix and ROC curve respectively.

5.5 Hybrid model(CNN + LSTM)

This model combines the strengths of CNN for local feature extraction and LSTM for sequence modeling. The CNN layer extracts key n-gram features, which are then passed

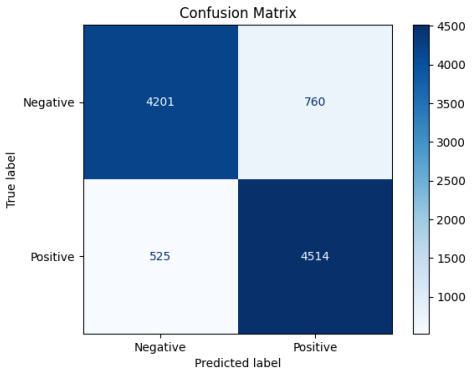

Fig. 10 Confusion matrix (Bi-LSTM (2-layer))

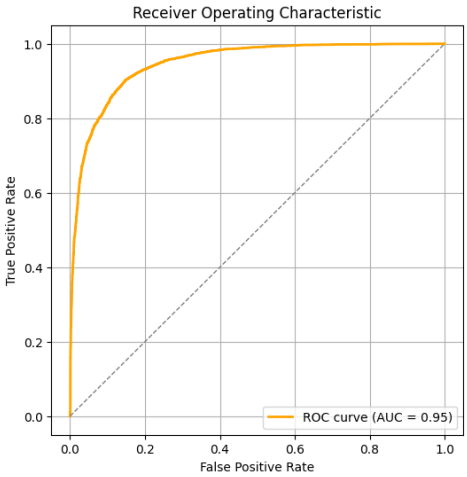

Fig. 11 ROC curve with AUC = 0.95 (Bi-LSTM (2-layer))

to an LSTM for sequential understanding. This hybrid model achieved a competitive accuracy of 88.30%, demonstrating its effectiveness in leveraging both local and global features in text. Figures 13 and 14 illustrate the performance of the Bi-LSTM model using the confusion matrix and ROC curve respectively.

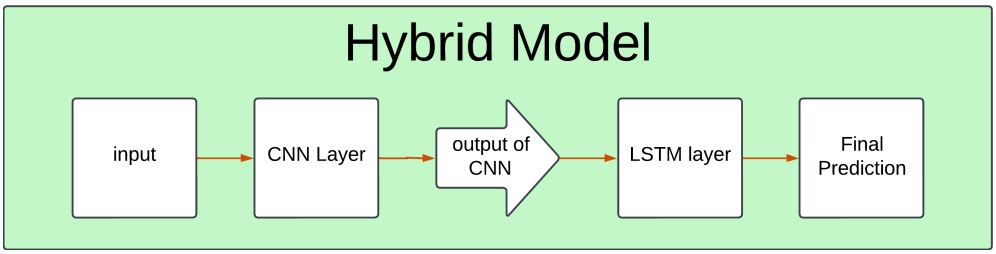

Fig. 12 Architecture of the hybrid CNN-LSTM model

6 Results

This section presents the performance comparison of the different deep learning models evaluated on the sentiment analysis task. All models were trained and tested under identical conditions to ensure a fair comparison, with consistent preprocessing, embedding strategies, and hyperparameter settings.

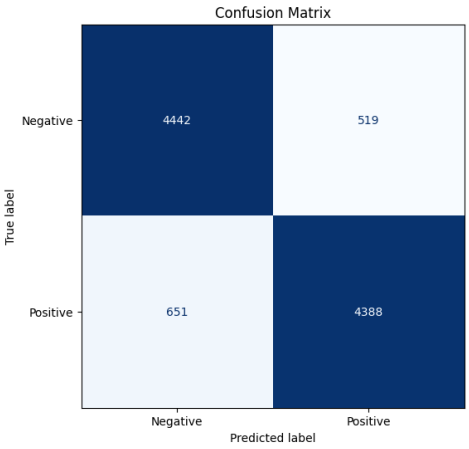

Fig. 13 Confusion matrix (hybrid model)

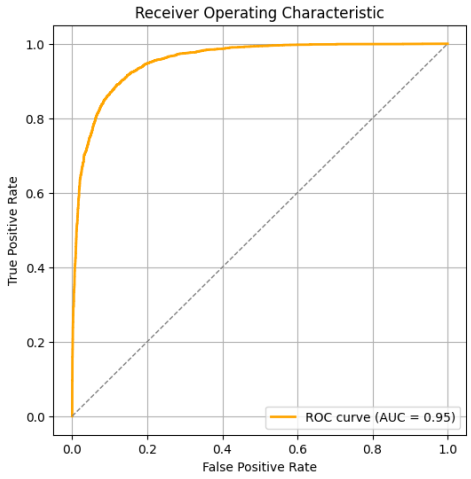

Fig. 14 ROC curve with AUC = 0.95 (hybrid model)

The primary metric used for evaluation was **classification accuracy** on the test dataset. The results highlight the distinct strengths and trade-offs of each architecture:

- **CNN**: Achieved the highest accuracy of **89.02%**, indicating its effectiveness in capturing local textual patterns and discriminative features in sentiment-labeled data.
- **Bi-LSTM (2-layer)**: Followed closely with an accuracy of **88.52%**, suggesting that deeper recurrent networks can better capture contextual dependencies compared to their shallower counterparts.
- **Hybrid CNN-LSTM**: Achieved a balanced performance with **88.30%** accuracy, successfully integrating local feature extraction with sequential modeling.
- **LSTM**: Performed moderately well with an accuracy of **88.05%**, effectively learning long-term dependencies but lacking the bidirectional or convolutional enhancements.
- **Bi-LSTM (1-layer)**: Recorded the lowest accuracy of **87.63%**, suggesting that bidirectional processing alone was not sufficient and may have introduced unnecessary complexity without improving generalization.

These results indicate that while complex architectures can offer better feature representation, simpler models like CNN can outperform them when appropriately tuned for specific tasks. The findings also demonstrate that combining architectural strengths, as in the hybrid model, can lead to robust and competitive performance.

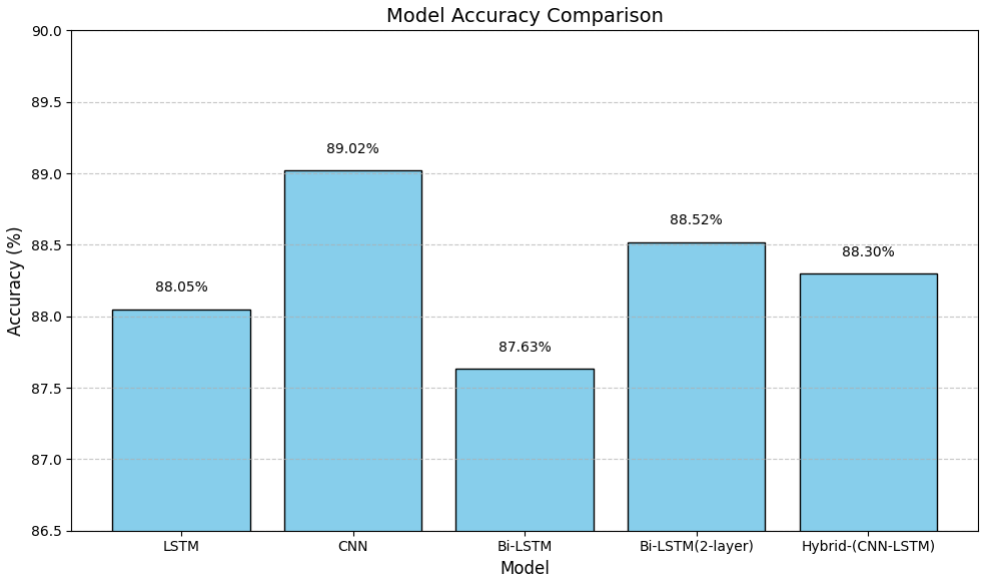

Fig. 15 Comparison of Model Accuracies

7 Conclusion

In this study, we explored and compared multiple deep learning architectures for sentiment analysis using a consistent experimental setup and the pre-trained `word2vec-google-news-300` embeddings. The models included LSTM, CNN, Bi-LSTM, Bi-LSTM (2-layer), and a Hybrid CNN-LSTM model. Each architecture was evaluated based on its ability to accurately classify movie reviews as positive or negative.

Our results demonstrate that convolutional neural networks (CNN) outperformed all other models with the highest test accuracy of **89.02%**, indicating their strong capability in capturing local dependencies and n-gram-like patterns in text data. The two-layer Bi-LSTM model also performed well, showcasing the importance of deep sequential modeling. The hybrid model, which combines CNN and LSTM, achieved a balanced accuracy of **88.30%**, validating the effectiveness of integrating both local and sequential features.

Overall, the performance of all models was relatively close, affirming the importance of model selection based on task requirements and resource constraints.

7.1 Limitations

While our research provides valuable insights, it has several limitations that should be acknowledged:

- The focus on binary sentiment classification (positive/negative) ignores the more nuanced sentiment spectrum that includes neutral, mixed, or varying degrees of positive/negative sentiment.
- The fixed sequence length of 300 tokens might truncate important information in longer reviews or introduce unnecessary padding in shorter ones.
- Our evaluation focused solely on the IMDB dataset, limiting the generalizability of findings to other domains or types of text.

7.2 Future Work

- Explore advanced embeddings like GloVe, FastText, and contextual models such as BERT for better context understanding.
- Develop hybrid architectures combining CNN, LSTM, or Transformer models to improve classification accuracy.
- Incorporate attention mechanisms to focus on key words or phrases, improving interpretability and accuracy.
- Optimize the model for real-time sentiment analysis on streaming social media or user feedback data.
- Develop explainability tools to visualize word importance and model decision-making processes.
- Expand the model to support multi-lingual sentiment analysis using multi-lingual embeddings or translation pipelines.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473, 2014.

[2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, *Enriching word vectors with subword information*, Transactions of the Association for Computational Linguistics, 5:135–146, 2017.

[3] S. Poria and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, pp. 42–49, Sep. 2016.

[4] K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, *Deep Learning*, Singapore: Springer, 2018, pp. 27–34.

[5] J. Einolander, "Deeper customer insight from NPS-questionnaires with text mining - comparison of machine, representation and deep learning models in Finnish language sentiment classification," 2019.

[6] P. Chitkara, A. Modi, P. Avvaru, S. Janghorbani, and M. Kapadia, "Topic spotting using hierarchical networks with self attention," Apr. 2019.

[7] F. Ortega Gallego, "Aspect-based sentiment analysis: a scalable system, a condition miner, and an evaluation dataset," Mar. 2019. [Online].*International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol. 9, no. 2/3, May 2019.

[8] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 1, Dec. 2015.

[9] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?" In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP '02*, 2002, vol. 10, pp. 79–86.

[10] A. Y. Ng, C. Potts, R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, and C. D. Manning, "Recursive deep models for semantic compositionality over a sentiment treebank," *PLOS One*, 2013.

[11] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," Unpublished manuscript.

[12] H. Cui, V. Mittal, and M. Datar, "Comparative experiments on sentiment classification for online product reviews," In *AAAI'06 Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.