Sentiment Analysis of IMBD Movie Reviews using LTSM Neural Networks

Akshat Baheti and **Kabiir Krishna**
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada
abaheti@uwaterloo.ca, k7krishna@uwaterloo.ca

Abstract

In the digital age, online reviews significantly influence decisions about movies and TV shows. This paper explores the use of Long Short-Term Memory (LSTM) neural networks for sentiment analysis of IMDB reviews. Using distilBERT and VADER, we generate continuous sentiment scores ranging from -1 to 1 for our training dataset. These scores train the LSTM model to handle the sequential nature of textual data, accurately identifying consensus and overall sentiment across reviews.

Our approach helps the entertainment industry understand audience preferences, guiding marketing strategies, recommendation systems, and content creation. Consumers benefit from the wisdom of the crowd, which helps them make better choices. The technology can also be extended to other areas such as product reviews and social media monitoring.

Experiments show that our model effectively captures and analyzes sentiment from large-scale data, demonstrating the potential of sentiment analysis to improve decision-making and tailor content to audience expectations. Github repository. (https://github.com/Kabiirk/MSCI641_project ; (incase embedded link doesn't work)
Keywords: NLP, LSTM, BOW (Bag of Words) VADER, DistilBERT, Tokenizer, Sentiment, GloVe.

1 Introduction

In today's world, the Internet plays a major role in our decisions, especially when it comes to entertainment. Before choosing a movie or TV show, we often look to online reviews and ratings to help us make our decisions. This trend has led to the rise of "video reviews" and in-depth articles, making it easier for us to choose content worth watching.(Bosques Palomo et al., 2024)

Think about planning a movie night or choosing a new TV to binge-watch. Wouldn't it be great to have an easy way to find out what others think instead of sifting through countless reviews and ratings? This is where sentiment analysis techniques in natural language processing (NLP) excel. They help us understand what people think and feel about movies and TV shows, allowing us to make better choices faster.

It's important for the sites and companies that provide these reviews to understand what their users think. It helps them recommend content that users will enjoy. Advanced methods like Long short-term memory (LSTM) neural networks are well-suited to this task. They are better than traditional methods because they can handle more complex data without losing accuracy. They are also very efficient and work well even on devices with limited resources.

The goal of this work is to make it easier for people to decide what to watch. We plan to collect user reviews and ratings from IMDB and use an LSTM model to analyze the opinions expressed in these reviews. By doing this, we can give each review an "emotional score" in the domain of [-1,1] that shows all the thoughts and feelings of the viewer. These insights will help users understand the public opinion on new movies and TV shows clearly and easily, making it easier to decide to watch the movie/show. (Bosques Palomo et al., 2024)

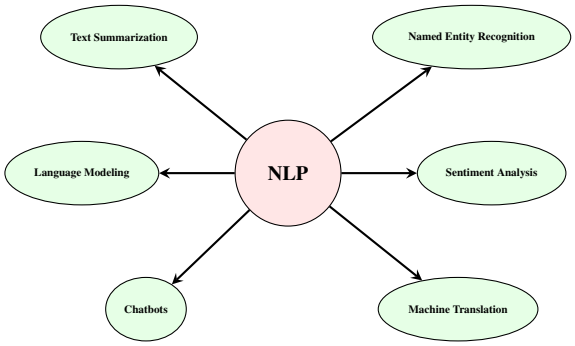

Figure 1: Applications of Natural Language Processing

2 Literature Survey

Advancements in sentiment analysis have seen significant innovations aimed at enhancing model accuracy and interpretability. A noteworthy contribution by **Rushlene Kaur Bakshi, Navneet Kaur, Ravneet Kaur, and Gurpreet Kaur**. explored the **impact of social media interactions on consumer opinions**, revealing how tweets from average individuals can shape public sentiment. This foundational work highlights the pivotal role of online interactions in influencing consumer behavior and sets the stage for subsequent research. (Bakshi et al., 2016)

Building on this understanding, **Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts**. advanced the field by developing a model that **combines unsupervised and supervised techniques** to generate word vectors with nuanced linguistic features. Their approach improves the alignment between string data and neural network models, offering a robust framework for more accurate sentiment analysis through sophisticated word embeddings. (Maas et al., 2011)

Furthermore, **Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek** enhanced the interpretability of **sentiment models by applying Layer-wise Relevance Propagation (LRP)** to a word-based bi-directional LSTM model. Their work demonstrates how LSTMs, when combined with LRP, can effectively perform emotion detection, making the model's predictions more transparent and reliable. (Arras et al., 2017)

In a novel approach, **Yukun Ma, Haiyun Peng, and Erik Cambria** addressed the complexities of targeted aspect-based sentiment analysis by embedding **commonsense knowledge into an attentive LSTM network**. Their use of hierarchical attention mechanisms significantly improves the accuracy and relevance of sentiment predictions, showcasing an advanced method for integrating external knowledge into LSTM models. (Ma et al., 2018)

The practical application of these models was showcased by **Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis**, who employed **deep LSTM networks with attention mechanisms** in their work for the SemEval-2017 Task on Twitter sentiment analysis. Their effective preprocessing techniques for unstructured tweet data highlight the potential of deep learning systems in handling diverse sentiment analysis tasks.

Adding a new dimension, **Alec Yenter, and Abhishek Verma** introduced a **hybrid model that combines CNN kernels with LSTM layers**. This innovative approach enhances sentiment analysis performance on IMDb reviews by capturing complex text patterns through a blend of CNN and LSTM architectures. (Yenter and Verma, 2017)

Collectively, these advancements provide a comprehensive understanding of sentiment analysis methodologies, from foundational insights to cutting-edge hybrid models. Each contribution builds on the previous ones, offering valuable tools and techniques for improving sentiment score prediction and model effectiveness.

3 Pre-requisite and Tools

3.1 Prerequisites

To undertake this project, having a foundational understanding of certain key areas will be beneficial. Familiarity with **Natural Language Processing (NLP)** concepts is advantageous, as it forms the core of analyzing and understanding human language data. Understanding basic neural network principles, especially **Long Short-Term Memory (LSTM) networks**, will help in comprehending how we train models to predict sentiment scores from text.

Additionally, knowledge of sentiment analysis techniques, such as **lexicon-based methods or like BOW (Bag-of-Words) approach**, will help assign sentiment scores to textual data.

3.2 Tools and Frameworks

- **Programming Language**: Python
- **Data preprocessing:** NumPy, Pandas, NLTK, regex (re), tokenizer, GloVe.
- **Data collection**: Beautiful Soup, Requests.
- **Training Data Generation**: VADER and DistilBERT.
- **Model Development**: Pytorch.
- **Data visualization**: Plotly Express.
- **Web Dashboard**: Streamlit

4 Methodology and Discussion

4.1 Data Collection and Cleaning

First, we collected reviews from IMDB using BeautifulSoup, a web scraping framework. ***BeautifulSoup*** traverses the HTML DOM of a particular

movie review page whose ID is given to extract reviews and their metadata, such as the title, author, date, stars (out of 10), and the review text. The extracted data initially contains noise, such as unnecessary whitespaces and special characters. Therefore, we performed data cleaning to remove these artifacts, convert text to lowercase, and ensure the date columns are in the appropriate datetime format with the help of ***Numpy, Pandas and Regex***. This preprocessing step ensures the data is in a usable state for further analysis.

4.2 Preparing Training Dataset

Although we had the data, we were missing the labels or sentiment scores for that data. Therefore, after cleaning the data, we utilized ***VADER (Valence Aware Dictionary and Sentiment Reasoner) and DistilBERT*** to generate the sentiment scores for training data. VADER assigns a compound score to each review, indicating its overall sentiment, ranging from -1 (most negative) to +1 (most positive). DistilBERT, a transformer model, was used to augment the VADER data. Since VADER is a 'weak' classifier, we used a strong classifier (DistilBERT) & averaged the scores from both. This resulted in a balanced and reliable sentiment score. These scores were used to create continuous output values and Polarity labels for the training dataset. We prepared a training dataset with over 24,000+ reviews from various movies to train our model.

4.3 Pre-processing Data

Before we can use our training data with the model, we need to transform it into a format the model can work with. First, we use the ***Tokenizer*** from the ***torchtext.data*** module to break the text into individual words or tokens. Then, we convert these tokens into numbers using a method called ***lookup_indices***, which maps each word to a unique index based on a dictionary we've created.

After that, we use ***GloVe*** to turn these numbers into word embeddings. GloVe creates these embeddings as vectors, which are numerical representations of words in a continuous vector space. They capture semantic meanings and relationships between words, allowing words with similar meanings to have similar vector representations. We then feed these embeddings into our model, which helps the model understand and learn from the text data. (**?**)

4.4 Choice of Neural Networks

4.4.1 Why Neural Networks for NLP ?

The fact that we don't need to manually create every feature since the network is intelligent enough to recognise features in each sentence on its own may be the primary benefit of using neural networks. As compared to conventional methods, neural networks provide us a more intuitive method of instruction and evaluation since it most closely resembles human behavior and, by extension, human reaction, which is what NLP accuracy is dependent on. (Arras et al., 2017)

4.4.2 Why LSTMs ?

Older models like Perceptron-Based ANNs struggle with sequential data, they are unable to store information necessary for a thorough processing of the texts. (Qaisar, 2020)

The majority of RNN's significant NLP issues stem from its Vanishing Gradient issue. Additionally, as time goes on and RNNs are fed an increasing amount of new data, the old data becomes diluted between the new input, the activation function transformation, and the weight multiplication, causing the RNNs to "forget" about the old data. This indicates that they have a decent short-term memory but a minor recall deficit for events that occurred some time ago. Consequently, a device with **LONG SHORT TERM MEMORY** is needed.(Arras et al., 2017)

LSTMs on the other hand, Have a GRU-based Unit, which help handle sequential data better than ANNs/RNNs. LSTMs handle the problem of vanishing gradient as well by using a unique additive gradient structure that includes direct access to the forget gate's activations. (Qaisar, 2020)

In short, LSTMs remember more important things for a longer time, by regulating what information is kept and propagated and what information is forgotten and discarded.

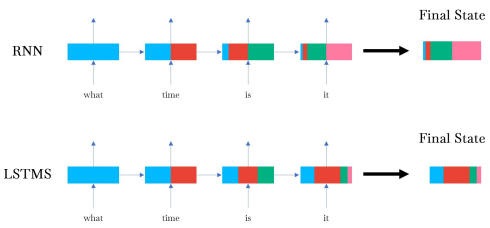

Figure 2: Comparison between LSTMs & RNNs retain sentence memory.

4.5 Model Building and Training

We developed the LSTM model using PyTorch. The model consists of the following components:

- **Embedding Layer**: Converts word indices into dense vectors. Padding is used to handle sequences of varying lengths.
- **Bidirectional LSTM Layer**: Processes the embedded vectors in both forward and backward directions to capture context from both ends of the sequences.
- **Dropout Layer**: Applied to prevent overfitting by randomly dropping units during training.
- **Fully Connected (Dense) Layer**: Generates the final sentiment score predictions from the LSTM outputs. For bidirectional LSTM, it combines outputs from both directions.

The model was trained using our dataset, and its performance was optimized over several epochs. Training and validation data were used to monitor and prevent overfitting, ensuring effective learning and reliable sentiment score predictions. (Qaisar, 2020)

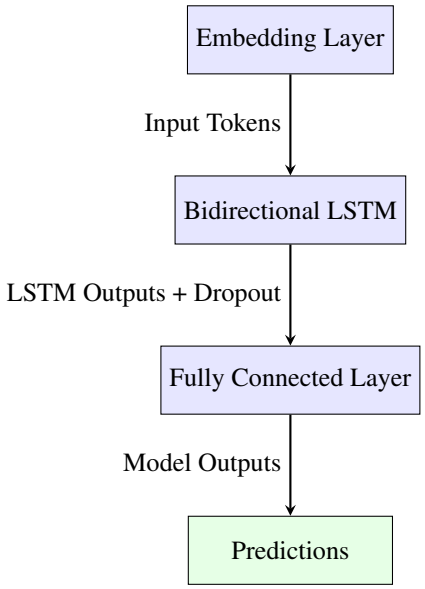

Figure 3: Architecture of the LSTM Model

4.6 Testing Phase

In this stage, we put our pre-trained LSTM model to work by predicting sentiment scores for reviews that we hadn't previously predicted with VADER. We start by converting each review into numerical vectors using the ***tokenizer.texts_to_sequences()*** method, which turns text into a sequence of numbers based on our vocabulary. To make sure all sequences are of the same length, we use ***pad_sequences()***.

Once the reviews are processed and ready, we input them into the LSTM model using the ***model.predict()*** method to get sentiment scores.

We found that, for a previously unscored dataset, the sentiment scores from our LSTM model closely match those from VADER and DistilBERT. This consistency shows that our LSTM model is effectively capturing sentiments in a way that's comparable to these well-established methods, confirming the accuracy and reliability of our approach.

4.7 Visualization

In this part of the project, we focus on visualizing the sentiment scores predicted by our model. Large datasets in raw tabular format can be overwhelming and hard to interpret. Visualization helps to make sense of the data and reveal important patterns.

We use **Plotly Express** for this task. Plotly Express is a high-level interface for creating interactive plots and visualizations in Python.

With these tools, we create various types of visualizations, such as ***line plots*** to show how sentiment trends change over time and ***bar charts*** to compare sentiment across different categories. These visualizations make it easier to understand the data and communicate our findings clearly.

By turning our data into clear and engaging visuals, we can better grasp the insights from our sentiment analysis and present them in a way that's easy to understand and impactful.

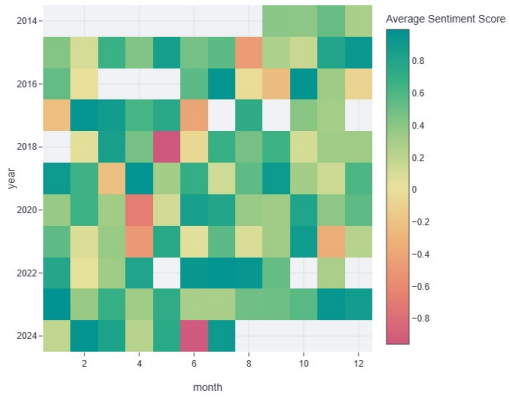

Figure 4: Heatmap of Sentiment scores of movie The Equalizer

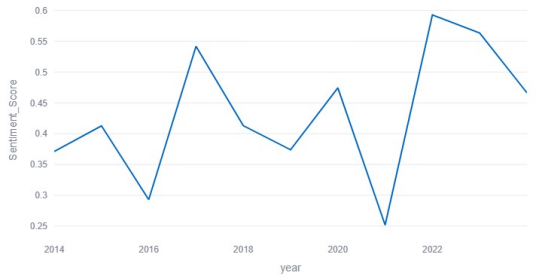

Figure 5: Line plot of Sentiment scores of movie The Equalizer

4.8 Model Deployment and Web Application

In the final phase, we enhanced the accessibility and functionality of our sentiment analysis model by deploying it with Streamlit, a framework designed for building interactive web applications. This deployment enabled us to develop a website where users can input the IMDB ID of any movie or TV show to engage with our sentiment analysis engine.

Upon entering the IMDB ID, the application performs several automated tasks. It first scrapes relevant reviews associated with the specified ID and then processes these reviews using our sentiment analysis model. The results are presented through a sophisticated and user-friendly interface.

The web application offers several key features:

- **Sentiment Analysis Visualization:** The application generates a line plot that illustrates sentiment trends over time for the selected movie or TV show. This visual representation allows users to observe changes in public opinion and sentiment.
- **Final Sentiment Summary:** The model computes an aggregate sentiment score, which summarizes the overall sentiment of the content. This score is accompanied by a recommendation, providing users with a clear indication of the content's sentiment.
- **Interactive Filters:** The application includes various filters that enable users to customize their analysis. Users can adjust the sentiment range and select specific years to analyze how sentiment has varied during those periods, offering a tailored and detailed exploration of the data.

This deployment ensures that our sentiment analysis model is both accessible and practical, delivering a comprehensive tool for evaluating movie and TV show sentiments in a formal and engaging manner.

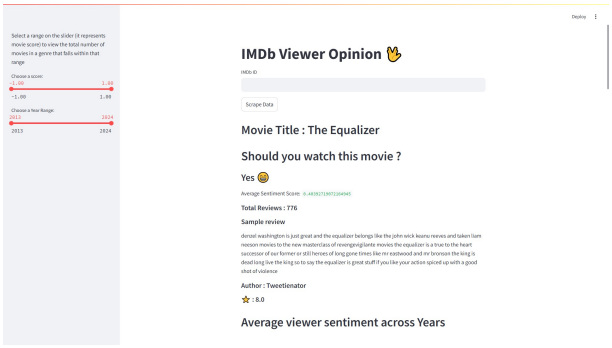

Figure 6: Streamlit Website

5 Results and Analysis

The performance of the LSTM model was evaluated based on various metrics, including loss, accuracy, precision, recall, and F1 score for both the training and validation datasets. The final metrics are as follows:

Table 1: Model Performance Metrics

Metric	Training	Validation
Loss	0.176	0.213
Accuracy	92.9%	92.0%
Precision	92.7%	91.2%
Recall	92.9%	92.0%
F1 Score	92.7%	91.0%

5.1 Analysis

Training Performance:

- The model achieved a training loss of 0.176, indicating that the model has effectively minimized the error during training.
- With a training accuracy of 92.9%, the model demonstrates strong performance, correctly classifying a high percentage of the training data.
- The precision, recall, and F1 scores of 92.7%, 92.9%, and 92.7%, respectively, reflect that the model performs well in distinguishing between classes and maintains a balance between false positives and false negatives.

Compared to our initial model submitted during the Project milestone, (trained on reviews scored only with VADER) and observed an increase in

Table 2: Model Performance on Validation Data

Metric	VADER	VADER + DistilBERT
Accuracy	89.4%	92.0%
Precision	88.1%	91.2%
Recall	88.4%	92.0%
F1 Score	88.2%	91.0%

model performance. The comparison is in Table 2:

Validation Performance (VADER + DistilBERT):

- The validation loss of 0.213 is slightly higher than the training loss, suggesting a small degree of overfitting. However, this increase is modest, indicating that the model generalizes well to unseen data.
- The validation accuracy of 92.0% is very close to the training accuracy, reinforcing the model's effectiveness on new data.
- Precision, recall, and F1 score values of 91.2%, 92.0%, and 91.0%, respectively, further support the model's robustness and its ability to maintain performance across different metrics.

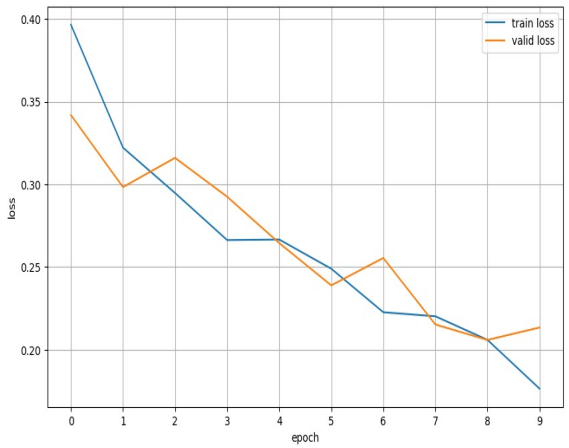

Figure 7: Training and Validation loss

5.2 Overall Assessment

The model's performance metrics are highly satisfactory, with only minor differences between training and validation results. The high accuracy, precision, recall, and F1 scores suggest that the LSTM model is well-tuned and capable of effectively predicting sentiment with a high degree of reliability. The slight discrepancy between training and validation metrics is typical and

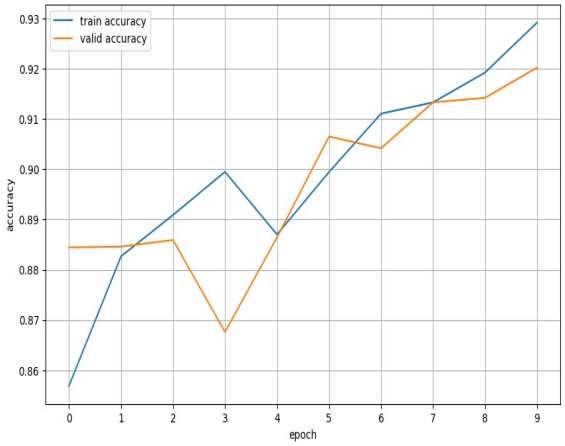

Figure 8: Training and Validation Accuracy

does not indicate significant overfitting, but rather demonstrates the model's robustness and ability to generalize.

6 Conclusion

This project explores the application of natural language processing (NLP) to analyze public sentiment through film reviews, demonstrating the advanced capabilities of NLP beyond traditional sales metrics. By leveraging VADER and DistilBERT, we constructed a comprehensive training dataset with over 30,000 observations, enhancing our sentiment analysis accuracy and providing a detailed understanding of audience opinions about movies and TV shows.

The deployment of our sentiment analysis model via Streamlit's web-based platform has facilitated user access to sentiment insights. Users can input an IMDB ID to view sentiment trends over time, obtain general sentiment summaries, and assess the overall worthiness of movies or TV shows. This user-friendly interface translates complex sentiment data into actionable information, making it practical for end-users.

7 Future Work

For future enhancements, the project could focus on several areas for improvement. Expanding the model's capability to detect a broader spectrum of emotional responses and experimenting with alternative neural network architectures could yield better accuracy. Additionally, addressing common issues such as spelling errors and keyword mismatches could refine the sentiment analysis results. The rapidly evolving field of NLP, including

innovations like Google's pQRNN model, presents new opportunities for enhancing our models and applications. This project lays the groundwork for future exploration and advancement in NLP applications.

Overall, this project illustrates how NLP can revolutionize the interpretation of public sentiment, offering valuable insights that support more informed decision-making for consumers and businesses. As technology advances, further opportunities to leverage NLP for enhancing various aspects of daily life will continue to emerge

8 Acknowledgement

We extend our sincere gratitude to Instructor Olga Vechtomova for her invaluable guidance and support throughout this project. Her insightful feedback and expertise were crucial in shaping our research and ensuring the success of our work. We also appreciate the assistance of Teaching Assistant Gaurav Sahu, whose technical support and advice significantly contributed to the project's development.

Additionally, we acknowledge the developers and maintainers of the tools and libraries utilized in this project, including VADER, DistilBERT, PyTorch, and Streamlit. Their open-source contributions provided the essential resources needed for our sentiment analysis model. We are also thankful to our peers and colleagues for their encouragement and constructive feedback, which helped refine our approach and achieve our objectives.

References

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 159–168, Copenhagen, Denmark. Association for Computational Linguistics.

Rushlene Kaur Bakshi, Navneet Kaur, Ravneet Kaur, and Gurpreet Kaur. 2016. Opinion mining and sentiment analysis. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 452–455.

Beatriz Bosques Palomo, Flor Velarde, Francisco Cantu-Ortiz, and Hector Ceballos. 2024. Sentiment analysis of imdb movie reviews using deep learning techniques. pages 421–434.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Saeed Mian Qaisar. 2020. Sentiment analysis of imdb movie reviews using long short-term memory. In *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, pages 1–4.

Alec Yenter and Abhishek Verma. 2017. Deep cnn-lstm with combined kernels from multiple branches for imdb review sentiment analysis. *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pages 540–546.