A Comparative Study Between Support Vector Machine and Long Short Term Memory Models on Sentiment Analysis of Movie Reviews

Pulkit Bhatt, Abhishek Deogam, and Neetu Gupta(✉)

Department of Computer Science and Engineering, Manipal University Jaipur, Jaipur, Rajasthan, India
{pulkit.219301408,abhishek.219301780}@muj.manipal.edu,
neetu.gupta@jaipur.manipal.edu

Abstract. Sentiment analysis has become an important aspect of natural language processing, particularly in evaluating public opinions and sentiments expressed in textual data. Reviews are a good source for critics and casual viewers to express how they feel about the movie. This research paper presents a comprehensive comparative study between Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks for sentiment analysis of movie reviews. Our main aim is to explore the strengths and weaknesses of these two approaches in capturing the nuanced sentiments embedded in movie-related text. Our study employs a diverse and well-curated dataset which consists of fifty-thousand movie reviews on IMDB, encompassing a wide range of genres and sentiments. The dataset is well balanced. The preprocessing involves techniques such as tokenization, stemming, and vectorization to ensure the models' effective comprehension of the semantic context.

Keywords: Sentiment Analysis · Text Classification · Natural Language Processing · SVM · LSTM

1 Introduction

The ability to analyze and interpret human sentiment has become increasingly important in today's era of digital communication and social media. Sentiment analysis is a natural language processing technique that is primarily concerned with binary text classification, such as opinion that is either positive or negative. Sentiment analysis is a branch of natural language processing that is essential to understanding the attitudes, feelings, and subjective opinions present in a wide range of textual material. Among the various applications of sentiment analysis, the evaluation of movie reviews holds great deal of significance, as it offers valuable insights into audience reactions and preferences within the ever-evolving landscape of the entertainment industry. It also serves as a great feedback mechanism which drives forward cinema as an art form. Support Vector Machines and Long Short-Term Memory networks are amongst few of the most commonly used

P. Pareek et al. (Eds.): IC4S 2024, LNICST 597, pp. 14–24, 2025.
https://doi.org/10.1007/978-3-031-77075-3_2

algorithms within the field of sentiment analysis. SVMs, known for their robustness and ability to handle high-dimensional data, have been extensively applied in sentiment analysis tasks. Their strength lies in identifying patterns and classifying data points based on their characteristics. They are well-suited for tasks with small to medium-sized datasets and provide a transparent decision boundary, which helps in interpretability. However, they may face limitations when dealing with sequential data, such as text, where understanding the context and order of words is crucial for accurate sentiment classification. LSTM networks are a type of recurrent neural network (RNN) which have become more and more popular for sentiment analysis in recent years. This is mainly because of their ability to capture long-range dependencies in text which makes them useful for tasks where context and order of words are of extreme importance (Long Short-Term Memory (LSTM) RNN Model - GM-RKB, n.d.). By processing sequences of words, they can effectively capture the nuances of sentiment expressed throughout a review. This capability has led to significant improvements in sentiment analysis accuracy compared to the old machine learning approaches. In our research, we will be developing sentiment analysis models from online movie reviews based on the above mentioned two main algorithms for classifying positive and negative reviews. We will compare the models' performance in terms of classification accuracy and also on the basis of the amount of time spent on model development and training. This research will help direct the selection of the most appropriate algorithm based on the particular requirements and constraints of a given task by outlining the advantages and disadvantages of these two popular text classification techniques. Furthermore, it contributes to the broader understanding of the trade-offs between different models in the sentiment analysis of reviews.

2 Literature Review

Recent years have seen a significant increase in interest in sentiment analysis due to its numerous applications in identifying consumer preferences, market trends, and public opinion. The viewers' opinions hold much significance in driving forward the entertainment industry. Due to the explosion in the amount of online data in recent years, it has become more and more significant. It can be used to understand public opinion, track product reviews, and monitor social media trends. In the context of movie reviews, it provides valuable insights into audience reactions and influences decision-making processes in the show business. In [1], the authors concluded that combining deep learning architectures with word-embeddings has outperformed theTerm Frequency Inverse Document Frequency- based models (TF-IDF). In [2], the authors have proven that the Random Forest algorithm would be the best algorithm to classify the IMDB dataset. The authors of [3] did direct word-embedding for small text regions, instead of using word vectors or even a bag of ngrams. Sentiment Analysis Techniques: Numerous methodologies have been employed for sentiment analysis, ranging from rule-based approaches to machine learning and deep learning techniques. Machine learning algorithms have proven effective in binary and multiclass sentiment classification tasks. SVM and its ability to find optimal hyperplanes for separating classes has been widely utilized in sentiment analysis [4].

Deep Learning in Sentiment Analysis: Sentiment analysis has been transformed by deep learning, with recurrent neural networks (RNNs) and their variant, LSTM, becoming more and more popular. They are designed to capture long-range dependencies in sequential data, and have shown remarkable success in tasks where context preservation is crucial, making them suitable for sentiment analysis in reviews with complex structures and nuanced sentiments [5].

Challenges in Movie Review Sentiment Analysis: Movie reviews pose unique challenges for sentiment analysis due to the presence of sarcasm, irony, and contextual complexities. Previous models may struggle to capture the subtle nuances in language, making it essential to explore the effectiveness of deep learning architectures [6].

Comparative Studies in Sentiment Analysis: Several comparative studies have explored the performance of various models in sentiment analysis tasks. In [7], the authors have found that a combination of Bag of Words combined with neural network gives us the model with the highest accuracy which is 82 percent accuracy and a training time around 1 min for classification of product reviews.

Hybrid Approaches: Recent research [7, 8] have focused on hybrid models that combine the strengths of machine learning and deep learning techniques.

Datasets for Movie Review Sentiment Analysis: The availability of diverse and well-curated datasets is fundamental to the success of sentiment analysis models. Previous studies have utilized datasets like the IMDb movie reviews dataset to benchmark and evaluate the performance of sentiment analysis models [9].

3 Methodology

3.1 Dataset

The IMDB 50 k movie reviews dataset is a widely used benchmark dataset in natural language processing and sentiment analysis. It consists of fifty-thousand movie reviews from the Internet Movie Database (IMDB), which are split into two sets: a training set with 25,000 reviews and a testing set with an additional 25,000 reviews. The reviews are labeled with binary sentiment values, indicating whether the sentiment expressed in the review is positive or negative. The dataset is typically balanced, meaning it contains an equal number of positive and negative reviews. The reviews in the dataset consist of textual content written by users expressing their opinions and sentiments about movies. The length of the reviews can vary, and they may contain both positive and negative sentiments. Sentiment analysis on this dataset presents challenges such as sarcasm, nuanced language, and context-dependent sentiment as shown in Fig. 1.

A typical review text would look something like this: Based on the tale of Anne Boleyn, a newly appointed lady-in-waiting to the Queen, this silent historical drama tells the passionate story of Henry VIII, the lustful King of England who enjoys hunting, feasting, drinking, being amused by his court jester, watching jousts, and chasing after young ladies who appear out of cakes and other pretty girls around the castle. He promptly dissolved his union, wed Anne, and instructed her to bear a male heir as it is her sacred duty. He quickly sets his sights on yet another lady-in-waiting once she falls short on that front. In contrast, Anne appears reluctant, agitated, or almost ready to cry for the majority of the movie.

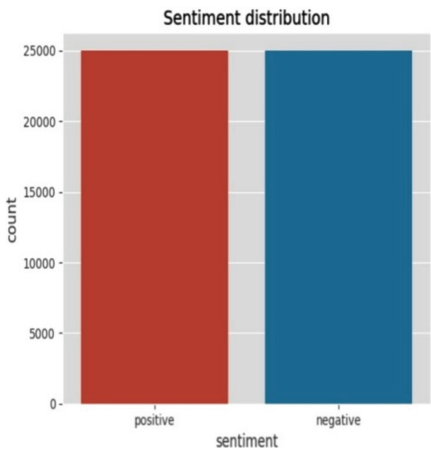

Fig. 1. Sentiment Distribution

She certainly doesn't seem like a content camper—or is she just behaving badly?·br /¿¡br /¿With a captivating tale that kept my attention for two hours, this film is a great source of entertainment. I really loved seeing the opulent medieval costumes displayed on a stunning sepia-toned print. In his superb depiction of King Henry the Eighth, Emil Jannings is incredibly stunning and unforgettable; it almost seems as though he WAS Henry the Eighth. I'm not sure I agree with the actress's portrayal of Anne; it felt a little too dramatic. This film's DVD has a fitting, well-executed piano soundtrack that is ideal for the narrative. A really good movie [10].

3.2 Data Preprocessing

Several techniques were employed in this study to complete the sentiment categorization job. As previously said, sentiment analysis involves a series of procedures that range from preprocessing and data preparation to data processing. Using the Pandas Python package, the dataset is first represented as a data frame. Next, the following procedures for data preparation areexecuted:

- Tokenization: Using word- embedding methods, all reviews are divided into individual words to create a vector of words. The Keras tokenizer is the one that is utilized with the remaining models..
- Removing Stop Words: Words that are frequently used in a language but are typically regarded as having little meaning or content value when it comes to text analysis are known as stopwords. In natural language processing (NLP) tasks, these words are frequently filtered out or excluded from text data during preprocessing. Stopwords are eliminated in order to concentrate on the more significant words that convey the text's primary semantic content. Stopwords are commonly used words that don't add much to the overall meaning of a text. The text's remaining words become more indicative of the main ideas of the content when these words are eliminated.

- Stemming: Stemming is a natural language processing (NLP) technique used to reduce words to their base form, which is often a common form shared by related words. The purpose of stemming is to simplify words and improve the efficiency of text processing and analysis by treating similar words as the same. In English, for example, the stem of words like "running," "runner," and "ran" is "run." Stemming algorithms aim to remove suffixes from words to extract their root form. While this process may result in the stem not being a valid word, it is acceptable for certain applications, such as information retrieval or text mining, where the focus is on the underlying meaning rather than grammatical correctness.
- Data Normalization and Cleaning: Applied to the entire dataset to ensure accurate interpretation. The goal is to eliminate stop words that can be shortened with the NLTK wordnet, HTML tags, and non-alphabetic characters. Moreover, repeated spaces are eliminated, even though punctuation marks might be crucial in certain NLP tasks, including sarcasm recognition..
- Data Encoding: Using real-valued vectors to represent words is one of deep learning's innovations. A variety of word embedding methods are available for text representation; they include the Bag of Words (BOW) using TF-IDF, Google's well-known Word2Vec model, Facebook's Fast Text representation, and Stanford University's GloVe. We employed BOW and GloVe in our investigation. The algorithm known as BOW, which extracts features from text, gets its name from the fact that it disregards word order. The TF-IDF approach uses BOW as a basis and attempts to give each word a weight that corresponds to its significance.

3.3 Exploratory Data Analysis

Upon exploring and analyzing the dataset, we found the most common words present in both positive as well as negative reviews. The word clouds for the same have been shown in Figs. 2 and 3.

The top frequently used words in positive as well as negative reviews have been shown in the graphs in Figs. 4 and 5 respectively.

4 Implementation

4.1 Support Vector Machine

Support Vector Machine works well for regression tasks as well as classification tasks. Its primary goal is to divide the datapoints of different classes represented as multi-dimension vectors into homogeneous groups by building a hyperplane, as illustrated in the Fig. 6 below, in order to classify a target object. Users must select the vector that best fits the given dataset among those produced by the kernel function in order to achieve the best performance.

4.2 Long Short Term Memory

A kind of recurrent neural network (RNN) architecture called Long Short-Term Memory (LSTM) was created to get around some of the drawbacks of conventional RNNs in

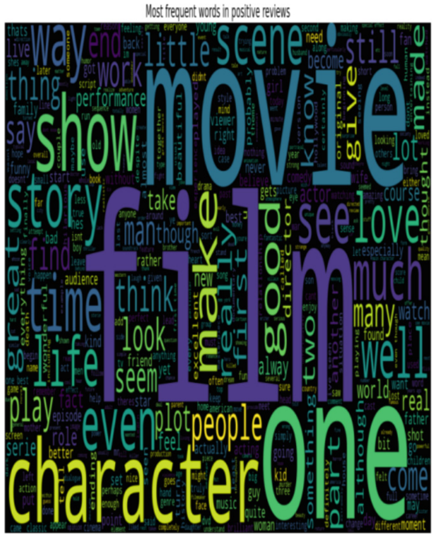

Fig. 2. Most Frequent Words in Positive Reviews

Fig. 3. Most Frequent Words in Negative Reviews

terms of identifying and understanding long-range relationships in sequential input. The memory cell, which enables the network to gather and store data for extended periods of time, is the essential component of an LSTM. Figure 7 illustrates how the Forget Gate decides whether data from the cell's prior state should be ignored.

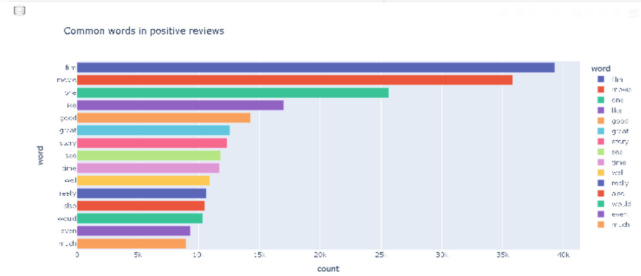

Fig. 4. Common Words in Positive Review

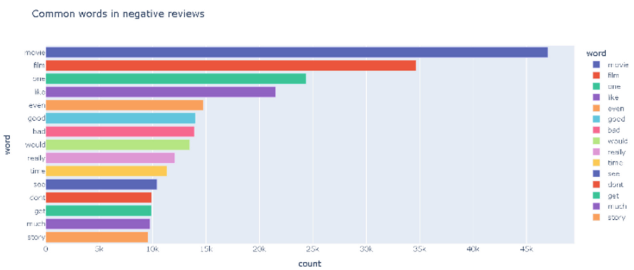

Fig. 5. Common Words in Negative Reviews

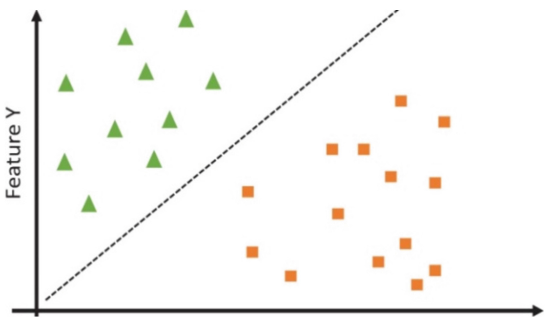

Fig. 6. Support Vector Classification

4.3 Feature Extraction

We used Word Embedding to convert our textual data to numeric form. Text must be converted to numbers in order for text data to be used as input into a neural network model. There are popular techniques like one-hot encoding which do a fair job in getting this done. Neural network models can be significantly impacted by passing sparse vectors of enormous sizes, in contrast to machine learning models. Consequently, we must transform our text into tiny, dense vectors [11]. We achieved this using Word Embedding. In the case of Word Embedding, every word is represented as an n-dimensional dense vector [11]. The strength of word embeddings is in the way it manages to retain

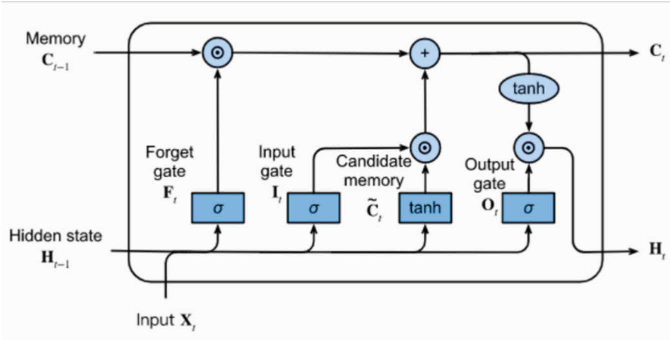

Fig. 7. Long Short Term Memory

the semantic relationship among words, as every token has n-dimensional parameters explaining its characteristics.

5 Results

We completed the model training stage after using the necessary preprocessing approaches. Thirty percent of our data was set aside for testing, while seventy percent was used for training. This separation ratio was chosen because the machine learning industry frequently uses it. There are an equal amount of reviews with positive and negative sentiment to balance the training and testing data.

5.1 Support Vector Classifier

The classification report for the Support Vector Classifier is shown in Fig. 8.

	precision	recall	f1-score	support
1	0.89	0.90	0.89	7513
2	0.90	0.88	0.89	7361
accuracy			0.89	14874
macro avg	0.89	0.89	0.89	14874
weighted avg	0.89	0.89	0.89	14874

Fig. 8. Classification Report for Support Vector Classifier

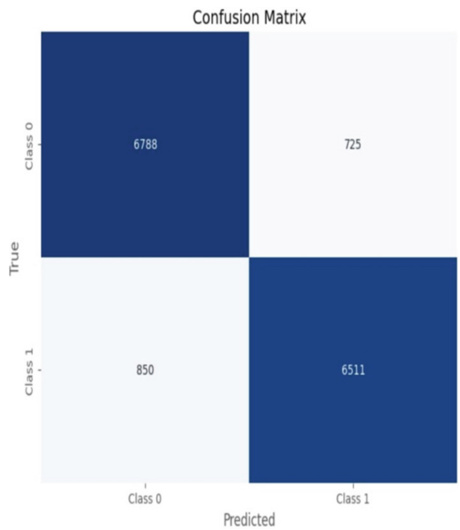

Fig. 9. Confusion Matrix for Support Vector Classifier

The accuracy came out to be 89.22 percent. Now we perform hyperparameter tuning to try and tweak the parameters of the model to achieve better accuracy. We implement this through a technique called Grid Search. Gridsearch is a hyperparameter tuning technique used to find the optimal combination of hyperparameter values for a machine learning model. Hyperparameters are parameters that are not learned during the training process but are set prior to training and can have a significant impact on the performance of the model [12]. There is a slight improvement in the accuracy which comes out to be 89.41 percent now. The confusion matrix for the same has been shown in Fig. 9. The classification report for SVC after Hyperparameter timing is shown in Fig. 10.

	precision	recall	f1-score	support
1	0.89	0.90	0.90	7513
2	0.90	0.88	0.89	7361
accuracy			0.89	14874
macro avg	0.89	0.89	0.89	14874
weighted avg	0.89	0.89	0.89	14874

Fig. 10. Classification Report for Support Vector Classifier after Hyperparameter Tuning

5.2 Long Short Term Memory

The accuracy of the LSTM classifier came out to be 86.29 percent. The confusion matrix for the same has been shown in Fig. 11 and the classification report in Fig. 12.

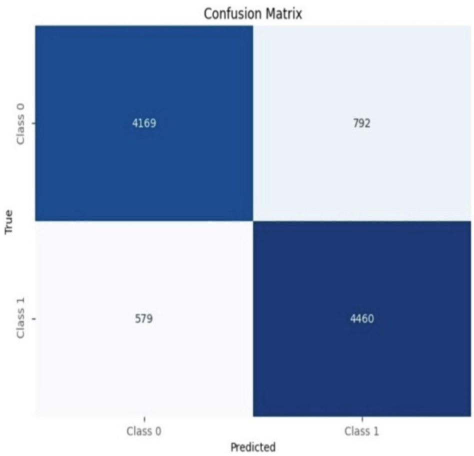

Fig. 11. LSTM Confusion Matrix

Classification Report:

	precision	recall	f1-score	support
0	0.88	0.84	0.86	4961
1	0.85	0.89	0.87	5039
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

Fig. 12. LSTM Classification Report

6 Conclusion and Future Scope

From the results above, we can conclude that the Support Vector Classifier outperforms the LSTM classifier by a considerable margin. Therefore, the Support Vector Classifier turns out to be a better choice in this context. This problem can be modelled as a multi-class classification problem in which we categorize the reviewer's sentiments in addition to positive and negative, such as "happy," "annoyed,", "amazed" and "impressed". This problem can be further reformulated as a regression problem, wherein instead of only predicting like or dislike for a given movie, we can also predict the degree of affinity (defined as rating).

References

1. Dang, N.C., Moreno-García, M.N., De la Prieta, F.,: Sentiment analysis based on deep learning: a comparative study. Electronics **9**(3), 483 (2020)
2. Yasen, M.: Movies reviews sentiment analysis and classification. IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (2019). https://doi.org/10.1109/JEEIT.2019.8717422
3. Johnson, R., Zhang, T.: Effective use of word order for text categorization with convolutional neural networks (2014). https://doi.org/10.48550/arXiv.1412.1058
4. Pang, B.L.: Thumbs up? Sentiment classification using machine learning techniques. Proceedings of EMNLP, pp. 79–86 (2002)
5. Tang, D., Qin, B., Feng, X., Liu, T.: Effective LSTMs for target-dependent sentiment classification (2015). https://doi.org/10.48550/arXiv.1512.01100
6. Mudinas, A., Zhang, D., Levene, M.: Bootstrap domain-specific sentiment classifiers from unlabeled corpora. Trans. Assoc. Comput. Linguist. **6**, 269–285 (2018)
7. Poomka, P.K.: Machine learning versus deep learning performances on the sentiment analysis of product reviews. Int. J. Mach. Learn. Comput. **11**(2), 103–109 (2021)
8. Birjali, M.K.-H.: A comprehensive survey on sentiment analysis: Approaches, challenges and trends. Knowl.-Based Syst. **226**, 107134 (2021)
9. Pouransari, H. and Saman G.: Deep learning for sentiment analysis of movie reviews **2**, 123–130 (2015)
10. Anne, S.: Review of Deception. Retrieved from IMDB (2006). https://www.imdb.com/review/rw1557506/
11. Malik, U.: Python for NLP: Word Embeddings for Deep Learning in Keras (n.d.). Retrieved from StackAbuse. https://stackabuse.com/pythonfor-nlp-word-embeddings-for-deep-learning-in-keras/
12. Tamanna Islam, A.B.: Prediction of breast cancer using traditional and ensemble technique: a machine learning approach. Int. J. Adv. Comput. Sci. Appl. **14**(6), 867–875 (2023)