Exploring Text Classification Methods for IMDB Dataset

Ahmed Jafer Osman Ahmed
dept. Computer Science and Electronic Engineering
University of Essex
Student ID: 2310485
eng.ahmedjafer@gmail.com

Abstract— **Recent years have exhibited a growing popularity in text classification in various domains, including business, marketing, entertainment, and government sectors. In order to achieve high performance, NLP researchers have investigated various algorithms and techniques. The study explored various machine learning techniques, such as support vector machine (SVM), logistic regression, and deep learning techniques, such as long short-term memory (LSTM), convolutional neural network (CNN), neural network (NN), and LSTM CNN as classifiers. The data was analysed and processed to ensure optimal performance. Many evaluation metrics were exploited to determine the effectiveness of the models. The results show that SVM and logistic regression outperformed deep learning models with and without stemming on most evaluation metrics.**

Keywords—Text Classification, Deep Learning, Sentiment Analysis, Machine Learning, Natural Language Processing

I. Introduction

Text classification is the process of analysing and classifying text into categories, and it’s considered one of the challenging topics in NLP. Text classification has been successfully applied to various domains, such as topic detection, spam e-mail filtering, SMS spam filtering, author identification, web page classification and sentiment analysis [1]. Sentiment analysis, also known as opinion analysis or opinion mining, has gained widespread acceptance in recent years, not just among researchers but also among businesses, governments, and organisations [2], especially with the rise of social media since these platforms became a way to share thoughts and opinions with a broad audience, by analysing social media posts, news articles, and online reviews. This provides valuable insight for governments to understand public concerns, researchers to identify trends and organisations to track their brand reputation.

IMDB is a website that contains a huge number of movie reviews as data. This data can be utilised to train different models to categorise new data and improve user experience.

II. Literature Review

Many studies have developed different approaches and models on IMDB dataset to achieve higher performance.

In [3], Tripathi et al. proposed different ML algorithms for classification: Logistic Regression, Naive Bayes, Decision Tree Classifier, and Random Forest Classifier. In this study, TF-IDF and counting methods were used for word embedding. Accuracy, Precision, Recall, F-score, and AUC were used to evaluate the performance of the algorithms. Although the study explored different algorithms, it did not explore more sophisticated word embedding techniques such as word2vec, Facebook’s FastText, and BERT.

In [4], S.M. Qaisar proposed a Long Short-Term Memory algorithm as a classifier for the IMDB dataset, utilising Doc2vec as a vectorization method known for its relatively low computational cost. The proposed architecture scored high accuracy, yet the study did not use different evaluation metrics to evaluate the system; the study did not explore alternative LSTM architecture and hypermeter configuration.

In [5], K. Amulya et al. proposed deep learning and machine learning algorithms for sentiment analysis. The study comprehensively compares the algorithms, offering valuable insights into their strengths and limitations.

III. Methodology

The study involves utilising various models on IMDB dataset followed by a comparative analysis of their performance using different evaluation metrics.

A. Data Preprocessing

The quality of data is crucial for achieving high performance. The data was processed using this Python package [6].

- HTML Tags Removal: The data is taken from the internet and contains HTML tags that have been removed as they are irrelevant to the review.
- Accented and special characters Removal: Accented and special characters (á è î ü $ @) have been removed since they add noise to the text and may affect the model's performance. In addition, non-alphabetical characters have been removed.
- Text Normalization: Transforming text into a single canonical form. For instance, words like "Bad" and "bad" are the same, but without the normalisation, they are different for the model. Therefore, the data was lowercased.
- Stopwords Removal: Words like "is," "the," "and" etc. do not contain much information about the text. Thus, it has been removed from the data, and only the meaningful words have been preserved.
- Stemming: It is a powerful technique that allows us to extract the core meaning of a word by removing its affixes. This allows us to obtain the word's root, minimising confusion between similar words with the same meaning. This study utilised stemming on the dataset, subsequently comparing the performance of the models with and without the application of stemming.

B. Data Exploration

Data exploration is vital in data processing since it gives insight and a deep understanding of the data’s characteristics, distribution, and patterns.

The first step was checking If the data was balanced. Having imbalanced data may cause the classifier to be biased toward one class because it does not have enough data to learn about the other classes, eventually leading to an accuracy paradox. Fig. 1 verifies that the data is balanced.

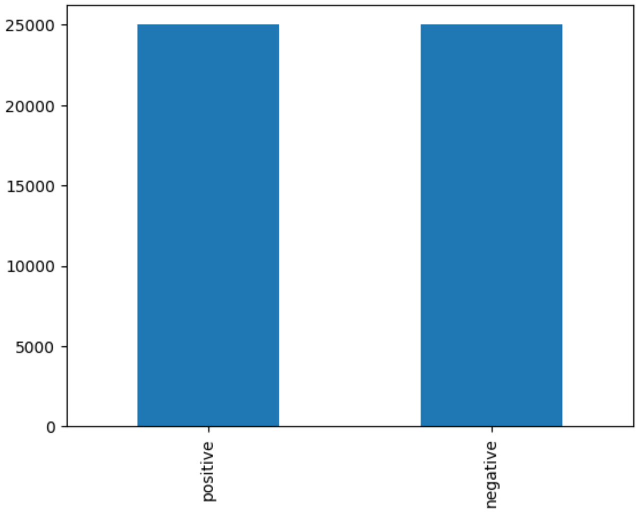

Fig.1. Classes Distribution Boxplot

Subsequently, numerical features were extracted from each review: the number of words, number of characters, number of stopwords, and average word length. A histogram was then plotted for each class for these features, as illustrated in Fig.2. It was observed that both classes had an equal distribution for each numerical feature.

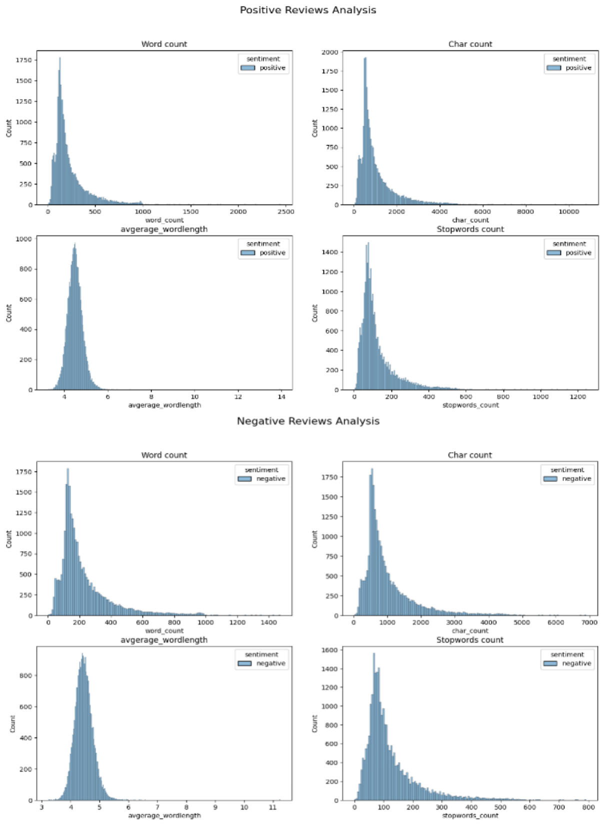

Fig. 2. Classes analysis

C. Word Embedding

Word embeddings are fixed-length vector representations for words. There are multiple ways to obtain such representations [7]. This study used two methods: TF-IDF and Word2Vec for word embedding.

Term Frequency-Inverse Document Frequency (TF-IDF) is used for support vector machine and logistic regression since these algorithms perform well with sparse vectors.

However, TF-IDF is computationally expensive for Neural Networks (NN), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and LSTM CNN. Therefore, pre-trained word2vec on Google News corpus was used since it reduces dimensionality and captures semantic relationships among words. This is because similar words have similar representations in the pre-trained model.

IV. Models

In this study, various classifiers were used to analyse different models extensively.

- Logistic Regression is a simple and interpretable model. It works well with sparse data like TF-IDF vectors, making it a suitable choice for this study.
- Support Vector Machine (SVM): Since SVMs use overfitting protection, they have the potential to handle large feature spaces. In addition, most text categorisation problems are linearly separable [8].
- Neural Network: Neural networks are considered one of the simplest deep learning models with high text classification accuracy.
- Long-Short Term Memory: LSTMs can learn long-range dependencies, do not suffer from vanishing gradient descent like RNN, and have shown promising results in text classification.
- Convolution Neural Network**:** CNNs are known for their use in computer vision but can. However, it can also be applied to text using one-dimensional convolutions. Due to their capacity to detect local features, CNNs are highly beneficial in text classification tasks.
- CNN LSTM: In [9], T. N. Sainathm et al. proposed a model that leverages CNN and LSTM. They used CNNs to reduce the spectral variation of the input feature and LSTM layers to perform temporal modelling.

V. Results

This section presents a summary of the results obtained from all the models.

TABLE I. RESULTS WITHOUT STEMMING

Model	**Without Stemming**			
	Accuracy	***Precision***	***Recall***	***F1 Score***
SVM	**89.39%**	**88.6%**	90.36%	**89.4%**
Logistic Regression	89.2%	88.1%	**90.5%**	89.3%
NN	85.25%	86.48%	83.56%	84.99%
CNN	85.37%	83.16%	88.7%	85.84%
LSTM	85.47%	83.91%	87.76%	85.79%
CNN LSTM	85.66%	85.27%	86.2%	85.73%

TABLE II. RESULTS WITH STEMMING

Model	Stemming			
	Accuracy	***Precision***	***Recall***	***F1 Score***
SVM	88.7%	**87.5%**	90.2%	88.8%
Logistic Regression	**88.71%**	87.2%	90.6%	**88.9%**
NN	81.47%	79.36%	85.06%	82.11%
CNN	81.57%	81.25%	82.08%	81.66%
LSTM	81.36%	79.55%	84.42%	81.91%
CNN LSTM	78.29%	72.04%	**92.44%**	80.98%

Table 1 summarises the results of all models without stemming. SVM outperformed all other models regarding all metrics except precision, where Logistic Regression surpassed it. Table 2 reveals that Logistic Regression performed better with stemming than other models in most metrics, except for Recall and precision, where logistic CNN LSTM and Logistic regression were superior.

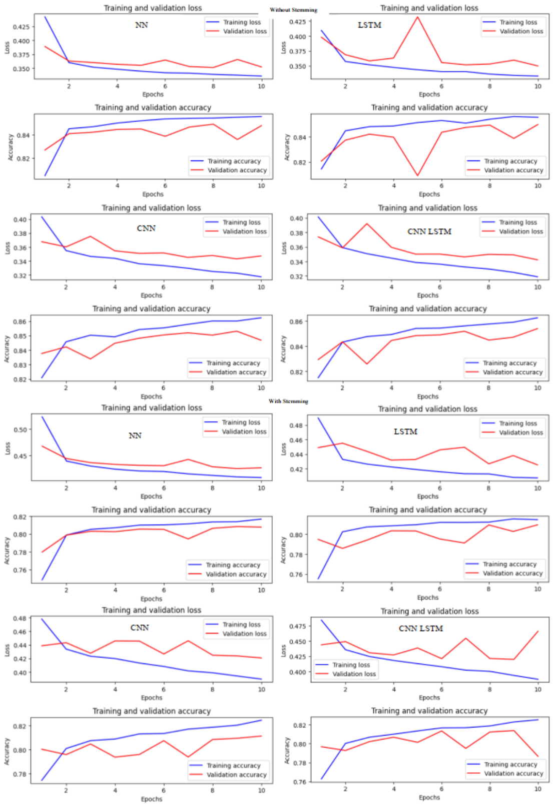

Fig. 3. Training and validation graphs

In both tables, deep learning models achieved good results, yet their scores were better without stemming the data because stemming led to the loss of some information, which decreased the performance. Fig. 3. Illustrates the training loss and validation loss for the models.

CONCLUSION

The study provided intensive exploration for text classification methods on the IMDB dataset. The data were preprocessed, and various word embedding techniques were used. Multiple models were trained and evaluated on multiple evaluation metrics. The study found that SVM and Logistic Regression outperformed deep learning algorithms on multiple evaluation metrics.

Although deep learning algorithms did not exceed the ML algorithms, experimenting with different architectures can improve it to achieve higher performance for each model. In addition, an attention mechanism can be added to LSTM to improve the model's ability to capture long-range dependencies and essential words in the input text. Finally, ensemble learning, which combines different model predictions, can increase the overall accuracy.

REFERENCES

[1] A. K. Uysal, and S. Gunal, "The impact of preprocessing on text classification," *Information processing & management*, vol. 50 no.1, pp. 104-112, 2014.

[2] M. Wankhade, A. C. S. Raoand C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges”, *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022, doi: 10.1007/s10462-022-10144-1.

[3] S. Tripathi, R. Mehrotra, V. Bansal and S. Upadhyay, "Analyzing Sentiment using IMDb Dataset," *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, Bhimtal, India, 2020, pp. 30-33, doi: 10.1109/CICN49253.2020.9242570.

[4] S. M. Qaisar, "Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory," *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, 2020, pp. 1-4, doi: 10.1109/ICCIS49240.2020.9257657.

[5] K. Amulya, S. B. Swathi, P. Kamakshi and Y. Bhavani, "Sentiment Analysis on IMDB Movie Reviews using Machine Learning and Deep Learning Algorithms," *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2022, pp. 814-819, doi: 10.1109/ICSSIT53264.2022.9716550.

[6] laxmimerit, preprocess_kgptalkie. GitHub [Online]. Available: https://github.com/laxmimerit/preprocess_kgptalkie

[7] F. Almeida, and G. Xexéo “Word embeddings: A survey,” *arXiv preprint arXiv:1901.09069*, 2019.

[8] T. Joachims, “Text categorization with Support Vector Machines: Learning with many relevant features”, in *Lecture Notes in Computer Science*, Lecture Notes in Computer Science, 1998, pp. 137–142. doi: 10.1007/bfb0026683.

[9] T. N. Sainath, O. Vinyals, A. Senior and H. Sak, "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 2015, pp. 4580-4584, doi: 10.1109/ICASSP.2015.7178838.