Interpretable Movie Review Analysis Using Machine Learning and Transformer Models Leveraging XAI

Conference Paper · January 2023

DOI: 10.1109/CSDE56538.2022.10089294

5 authors, including:

Farzad Ahmed
Ahsanullah University of Science & Tech
8 PUBLICATIONS **80** CITATIONS
SEE PROFILE

Samiha Sultana
BRAC University
5 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Md Tanzim Reza
BRAC University
46 PUBLICATIONS **97** CITATIONS
SEE PROFILE

Sajib Kumar Saha Joy
Ahsanullah University of Science & Tech
10 PUBLICATIONS **81** CITATIONS
SEE PROFILE

Interpretable Movie Review Analysis Using Machine Learning and Transformer Models Leveraging XAI

Farzad Ahmed[1], Samiha Sultana[2], Md Tanzim Reza[3], Sajib Kumar Saha joy[4], and Md. Golam Rabiul Alam[5]
[1,4] Department of CSE, Ahsanullah University of Science and Technology, Dhaka, Bangladesh
[2, 3, 5] Department of CSE, Brac University, Dhaka, Bangladesh
Email: [1]farzadahmed6@gmail.com, [2]samiha.sultana@g.bracu.ac.bd, [3]rezatanzim@gmail.com,
[4]joyjft@gmail.com, [5]rabiul.alam@bracu.ac.bd

***Abstract*—Text classification has been a common topic of interest for many years. A lot of advanced models has been developed so far in this area. But it is very difficult to understand how the models behave while predicting the class of the text. In our work, we utilized some models to classify the sentiment of movie reviews from text data and observed how the models behaved using Explainable Artificial Intelligence (XAI). At first dataset was collected and pre-processed. Then the processed dataset was separated into different train and test sets. The train set was used to classify using different different machine learning and neural network based models. The test set was used after training to evaluate the trained classifiers. Finally, the performance of the classifiers were compared and evaluated. After different variations of pre-processing and training steps, the best accuracy score of 91% was obtained using Roberta LSTM model. In the trained models, we sent texts that are correctly classified by RoBERTa models but misclassified by other models. Finally we figured out the reasons of misclassification with the help of LIME Algorithm.**

***Index Terms*—Natural Language Processing (NLP), BERT, Movie Review, Explainable AI**

I. Introduction

In our busy lives, we keep looking for a way to refresh ourselves from the monotony of life. In the late 18th century, the entertainment was more informal and non-commercial. For example, people used to travel to watch clowns for entertainment. Since the end of 19th century, cinema commercialized the entertainment industry [1]. Due to the advent of the internet, many people spend their free time watching movies. There are many genres in movies which include romance, comedy, thriller, horror, science fiction, fantasy, adventure, crime and so on [2]. The preferred genre can vary from person to person [3]. Comedy is the most popular genre [4] with 91% female and 90% male preferring it [5]. With the increasing size of the movie industry, the number of quality movies are also decreasing and so is peoples' will to enjoy movies by going to movie halls. According to a survey [6] about movies, 78 percent of respondents prefer to watch movies at home instead of going to a movie theater. Therefore, it is vital to find a suitable way of determining which movie to invest time in.

The movie rating and the emotion after watching the review does not match as many give a neutral score thus lowering the average score. Thus, a way to analyze the written review to provide accurate scoring is essential to differentiate the quality between movies. However, it is difficult to interpret whether a review is positive or negative. Therefore, we applied few algorithms on the same dataset to find the best way to predict movie by analyzing the reviews. In this paper, we use Logistic Regression(LR), Decision Tree(DT), Multilayer Perceptron(MLP), Bidirectional Encoder Representations from Transformers(BERT) with (long short-term memory networks) LSTM, BERT without LSTM, Robustly Optimized BERT Pre-training Approach(RoBERTa) LSTM to classify whether reviews are positive or negative. Additionally, we use Local Interpretable Model-agnostic Explanations(LIME) [7] to explain the results. Lastly we have mentioned how future improvement can be done.

II. Background Study

A. Previous Works

Movie Reviews have been an important sector for Natural Language processing for many years. In [8] the authors performed a sentiment based classification based on movie reviews that were web scraped from the internet. They used both supervised and unsupervised approaches to analyze the reviews. They obtained 85.54% accuracy for the supervised approach and about 77% in their unsupervised approach named Semantic Orientation. As their study took place in 2005 they did not have the access to enough data and also could not use advanced Natural Language approaches to get better results.

In [9] Zhuan and et al. a many learning based approach was built which used statistical analysis, movie knowledge and WordNet to summarize and classify movies whether they are good or bad. They used a total of 880 reviews to test system. One fifth of the data was used to test the model the rest of the data was used for training.They generated an average of 48.3% precision, 58.5% recall and 52.9% F1-score which outperfomed the approach of Hu and Liu in [10]. Again the

problem of these studies are lack of sufficient data and the advanced deep learning approaches used today.

In recent years, [11] , [12] and [13] conducted research on sentiment analysis on Movie Review. These studies focused on Supervised learning algorithms such as Bernoulli Naive Bayes, SVM, Decision Tree and Maximum Entropy approach. But these approaches could not get a score above 70% accuracy.

Recurrent neural networks (RNNs) can analyse sequences of any length, which gives them an advantage in text sentiment analysis applications. RNN, unfortunately, encounters a significant issue during training. Its vanishing gradient problem [14] [15] arises while processing lengthy sequences. It is challenging to grasp; the connection of lengthy sequences of RNN models which can cause gradient expansion or reduction in size of the training process. Luckily, the LSTM design [16] tackles this difficulty of learning long-term dependencies, which is done by integrating memory cells that can hold state for extended periods of time. In [17] S Anbukkarasi and S Varadhaganapath used self-collected Tamil tweets which were analysed for sentiment using a BiLSTM for character approach. There are a total of 1500 tweets in the dataset, evenly split between the good, negative, and neutral categories. Firstly, the removal of unimportant symbols, special characters and numbers were done in the text. Word embeddings of DBLSTM, based on the Word2Vec pre-trained model, were then used to represent the cleaned data. The dataset was split such that 80% was used for training and 20% was used for testing.The experimental results showed that the DBLSTM approach has an accuracy of only 86.2%.

In [18] the authors used BERT for Sentiment Analysis in Movies on IMDb dataset of over 5000 reviews and achieved accuracy of 92.28%. On the other hand, Chaturvedi and et al. used Bayesian networks and fuzzy recurrent neural networks [19] to classify texts whether they are factual or neutral. They achived 89% accuaracy on the TASS dataset.

In [20] IMDb, Sentiment140, and the Twitter US Airline Sentiment dataset are three sentiment analysis datasets that are used. This paper uses RoBERTa-LSTM model to produce experimental findings that performs well by obtaining precision 90%, 93% and 91% on the Sentiment140 dataset, IMDb dataset, and Twitter US Airline Sentiment dataset, respectively. However, this study did not show an explanation for the models.

In [21], authors got promising result on COVID-19 fake news detection task. They obtained 98% F1 score by using RoBERTa-LSTM model. In this paper also no explanations of the models were shown.

Explaining the results of text classification algorithms is an issue in Natural Language Processing . Liu et al. [22] proposed a explanation skeleton for text classification where the the model provides fine-grained explanation behind its decision. They peformed their testing using the PCMag and the Skytrax User Reviews dataset and showed that using the General Explanation Framework (GEF) with the baseline models (i.e., LSTM and CNN) improved the accuracy while providing necessary explanations.

The aforementioned methods performed well. Our intention is to propose a model that further improves the results that are showcased in the previous works and also explain the models prediction at the same time.

B. Dataset Details

The dataset used in our model is Sentiment Analysis on Movie Reviews which was taken from kaggle website : www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data.

Three unique csv files were available on the website: train, test, and sampleSubmission. Only the train.csv file was used, and the train data was divided into 80% train data, 10% validation data 10% test data. After a few tweaks, the dataset we used has four columns: PhraseId, SentenceId, text, and target.The file contains a total of 156060 data points. The labels in the target column are 0,1,2,3,4. 0 indicates a negative value, 1 indicates a somewhat negative value, 2 indicates a neutral value, 3 indicates a moderately positive value, and 4 indicates a positive value. There are too many 2 values in the collection. The dataset's average score is 2.06. As a result, it was dealt with during the preprocessing stage.The phrases in the text column come from the Rotten Tomatoes dataset. Each row has a phraseId and also has a sentenceId to make it easy to see which phrases belong to a single sentence.

III. ALGORITHMS

A. Feature Extractor

N-gram: N-grams are the fundamental features used in sequence-based sentiment analysis. An n-gram is an adjacent of n items from a given sample of text that we can pass to a model. This model can store the spatial information of the texts. Out of many forms, three forms are mainly used in N-grams:

- Unigram: An n-gram consisting of a single item from a sequence. Here n=1.
- Bigram: Bigrams are a special case of the n-gram where n is 2.
- Trigram: Trigrams are a special case of the n-gram, where n is 3.

Term Frequency-Inverse Document Frequency (TF-IDF): Moreover machine learning classifiers are built to work with numerical values instead of text data, so we have converted the text data into numerical values before passing it to the classifiers. For this conversion Count Vectorizer is used, which finds out all the unique words and for each row of text, it assigns a vector containing columns equal to the unique words and assigns count of specific words in that row of text in each column.

TF-IDF is another step of vectorization where it gives weight to each words depending on the importance of words in the sentence, which increases proportionally to the words in a sentence and is offset by the frequency of the word in the whole text. The equation for TF-IDF is illustrated below:

$$TF(t) = \frac{NumberOfTimesTermtAppearsInADocument}{TotalNumberOfTermsInTheDocument} \quad (1)$$

$$IDF(t) = \log_e \frac{TotalNumberOfDocuments}{NumberOfDocumentsWithTermtInIt} \quad (2)$$

$$TF - IDF(t) = TF(t) \times IDF(t) \quad (3)$$

N-gram conversion and TF-IDF vectorization was mainly necessary for the traditional machine learning classifiers. We experimented with a combination of these features. For the transformers, there are own dedicated dynamic pre-processing tools.

B. Classifier

For our proposed research, we used BERT and its variant RoBERTa for transformer models. On the other hand, for the traditional machine learning models, we used Logistic Regression (LR), Multi-layer Perceptron (MLP), and Decision Tree (DT) models. For both BERT and RoBERTa, we experimented by concatenating the features produced by LSTM layer with the produced features.

Decision tree (DT): DT is a type of classifier that classifies data using a model in the structure of tree [23]. DT splits the data into smaller subgroups eventually making a decision tree. The tree consists of decision nodes and leaf nodes, where the leaf nodes serve as the classification. DT uses entropy to determine the similarity of a sample. The formula of entropy is given below:

$$E(S) = \sum_{i=1}^{c} -p_i log_2 p_i \quad (4)$$

Multi Layer Perceptron (MLP): This classifier is an algorithm of supervised binary or multi-label classification [24]. In the training stage, the neurons learn and process the features one at a time. MLP extracts a linear decision boundary by learning the weights of the input features. The decision boundary separates the positive and negative data.

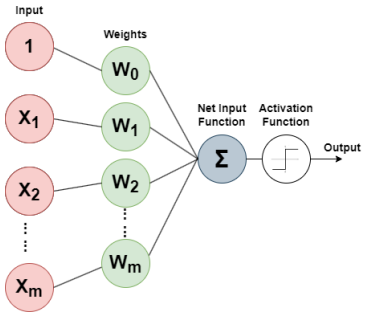

Fig. 1. Multi Layer Perceptron algorithm

Logistic Regression (LR): General regression is used to predict continuous values. Meanwhile, logistic regression is a variant of it that is used for classifying a discreet number of classes. The conversion from regular regression to logistic regression is done through the incorporation of sigmoid function. [25]. The equation for the sigmoid function is given below:

$$\sigma = \frac{1}{(1 + e^{-value})} \quad (5)$$

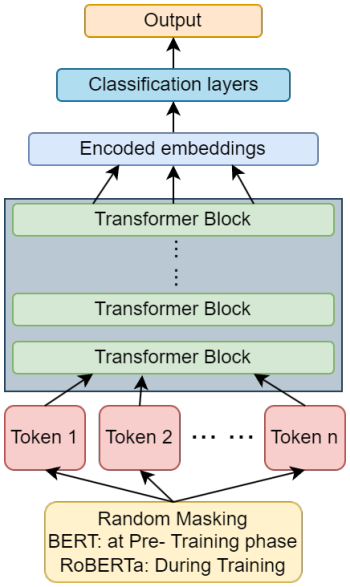

Fig. 2. General Flow of BERT and RoBERTa

BERT: BERT stands for Bidirectional Encoder Representations from Transformers. Just like other transformer models, BERT uses an encoder mechanism to learn language representation and express them as vectorized embeddings in traditional NLP tasks. [26] However, instead of reading a language sequentially either from right to left or from left to right like the standard transformers, BERT looks at part of the language through a Masked Language Model (MLM) approach and Next Sentence Prediction (NSP) approach simultaneously for a training phase. Through MLM, during a pre-training phase, randomly sampled parts of language are taken and few tokens are randomly masked from that. During training phase, BERT tries to determine the vocabulary ID of the masked token based on only the sampled context around it. Meanwhile, in NSP approach, BERT takes two sentences and tries to determine if the later sentence follows the former sentence, in a way similar to binary classification approach. This non-unidirectional style of learning by BERT helps to generate more effective contextual meaning of word through taking both left and right context into consideration. Once the BERT model is pre-trained, it can be further fine-tuned for more specialized tasks such as classification, question-answering, and so on.

RoBERTa: RoBERTa, on the other hand, is a BERT variant that can have a more robust learning of language representation through an improved MLM approach. As discussed, during training, BERT utilizes a static collection of masks already created in the pre-training phase. In contrast, RoBERTa dynamically generates the masks during the actual training phase, resulting in more variations of masking patterns. RoBERTa also generally uses longer sentence sequences, letting it take more context into consideration.

In the hyper-parameter tuning phase of the transformer based models, we have added a dropout layer with a rate of 0.2

to reduce overfitting problem. Tensorflow 2.2 library is used to implement these models. As optimizer we have used Adam [27] with a learning rate of 0.00002. Number of neurons in for the hidden layers and LSTM layers are tuned under manual observation on the validation data.

IV. Proposed Method

At first, dataset was collected and we performed some pre-processing on the dataset so that we can utilize it to train the traditional machine learning models.

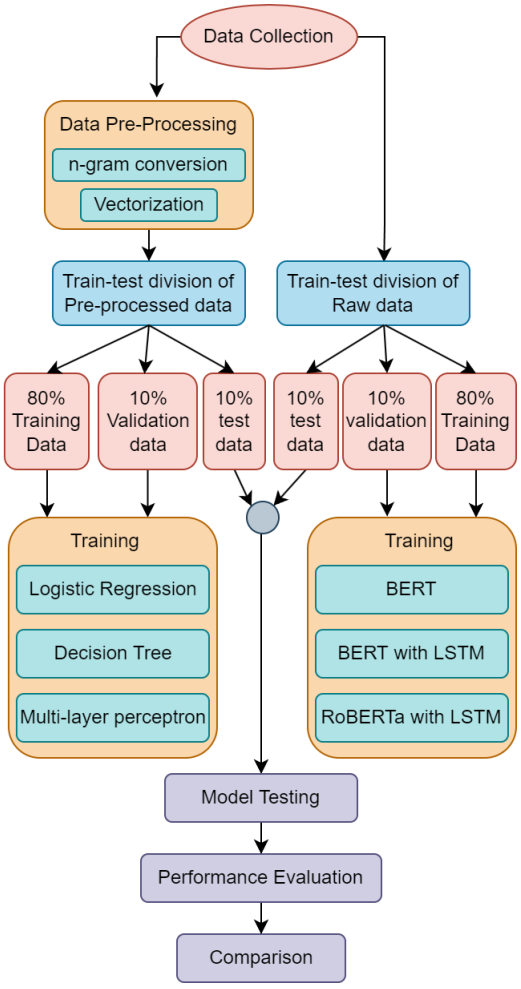

Fig. 3. Structure of the proposed system

For the first step, the entire dataset was tokenized in order to convert the sentences into tokens. Afterwards, all the capital letters were converted into small letters to achieve case independent classification. In order to reduce further variance of the dataset, we applied stemming and lemmatization to convert all the inflected forms of words to their root one. After removing all the anomalies in the text, we performed vectorization of the text that converted the tokens into numbers and also performed n-gram conversion (n=3) so that the traditional ML classifiers can take context into account, instead of treating each word token as individual units.

Finally, when the base dataset was ready, we separated the dataset into train, validation, and test sets. 80% of the entire dataset went to the train set, 10% from rest of the dataset went to the validation set, and the rest of the 10% samples went to the test set. In order to maintain the same ratio of postive and negative labels in both the train and test sets as the original dataset, we performed stratified split for train-test set conversion.

At this stage of the procedure, the dataset was ready for training. We decided to used LR, DT, MLP, BERT, and RoBERTa transformer. However, the traditional machine learning classifiers such as LR, DT, MLP are designed to work with numeric values so it is not possible for them to work directly with text data. Therefore, we had to convert the text data into numeric values through some vectorization process. For our proposed model, before training and performing classification with LR, DR, and MLP, we converted the dataset into different types of n-gram substrings. In this particular case, we converted tri-gram substrings. In addition to that, we used word vectorization and Term Frequency Inverse Document Frequency (TF-IDF) transformers in order to convert the text data into numeric values. On the other hand, in the case of the transformers such as BERT and RoBERTa, we passed the dataset through dedicated series of transformers in order to convert it into numerical vector embeddings. These vector embeddings were passed through Neural Network/LSTM later on for classification.

After processing, all the data were passed through the mentioned machine learning classifiers and trained using the train set. The validation set was used to tune the hyperparameters according to the training performance. Finally, we used the test set to evaluate the performance of the classifier. Since the transformer models generally require decent hardware for proper training, we used a machine consisting of Ryzen 3700X processor, 16 GB ram, and RTX 3070 GPU for training BERT and RoBERTa. Finally, all the results were compared and a conclusion was derived.

V. Result & Analysis

The results given by different models are summarized in figure 4. The RoBERTa LSTM model clearly outperforms all other models, with accuracy 0.91, precision 0.94, recall 0.89, and F1 score 0.91. To get an understanding about why Roberta LSTM model is performing the best, we take the help of Explainable artificial intelligence (XAI).

Model Name	Accuracy	Precision	Recall	F1 Score
BERT LSTM	0.84	0.86	0.82	0.84
BERT without LSTM	0.77	0.79	0.77	0.78
RoBERTa LSTM	**0.91**	**0.94**	**0.89**	**0.91**
Logistic Regression	0.82	0.83	0.87	0.85
Decision Tree	0.83	0.87	0.85	0.86
Multi Layer Perceptron	0.81	0.89	0.87	0.88

Fig. 4. Result obtained from different model

Explainable AI using LIME: Machine Learning Models are like a black box to normal users but with the help of Explainable Artificial Intelligence (XAI), [28] which is a assemblage of techniques and procedures, users can have an insight of the results that are produced by these models. Local Interpretable Model-agnostic Explanations (LIME) [7] model can demonstrate particular predictions of any classifier or regressor in a meaningful and intelligible way, by estimating them locally with an observable model. We have used this algorithm to understand how a review is classified as positive or negative.

Now lets consider two test samples one positive and one negative, that are misclassified by BERT LSTM but correctly classified by RoBERTa LSTM model. We are making this analysis relative to RoBERTa LSTM and BERT LSTM model as these two models give us the best results in all evaluation metrics.

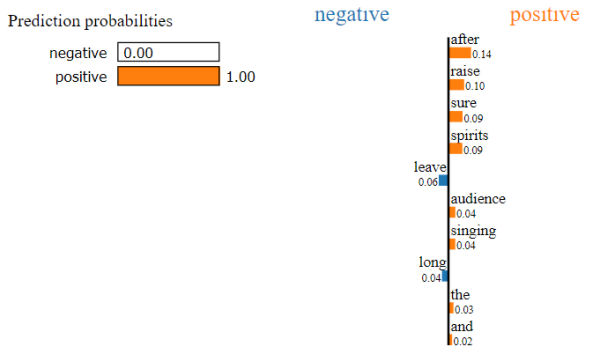

Fig. 5. Positive Review Correctly Classified by Roberta LSTM Model

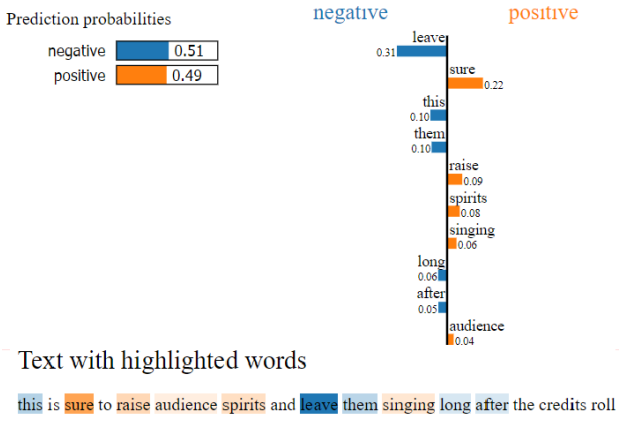

Fig. 6. Positive Review Misclassified by Bert LSTM Model

Figure 5 gives us an intuition about how the Roberta LSTM model labels the text "this is sure to raise audience spirits and leave them singing long after the credits roll" as positive review. We can observe that the prediction probabilities mentioned in the left most part is the local prediction score done by the explanation model. The text features and the values given in the middle part are the salient features and their coefficients. Positive words are on the right side of the line, while negative words are on the left. In the bottom section, the words of the text is highlighted according to the class and the color intensity is determined by the coefficient value of the words. We can see that 'sure', 'raise', 'spirits' and 'after' are the most dominating features that causes this text to be labeled as positive. Generally we can see that the overall contribution of positive words are noticeable compared to negative words. Thus the classifier model labeled this text as negative. On the other hand Figure 6 illustrates how Bert LSTM Model misclassifies the same positive review as negative. Here we can see that the model is hallucinating as the local prediction is 51% negative and 49% positive. From the color intensity we can see that this model gives the word 'leave' the most weight as it failed to capture the context of the given word. In this sentence the word leave is used in a positive sense 'leave them singing' but the model considers this review as negative without understanding the context of the review.

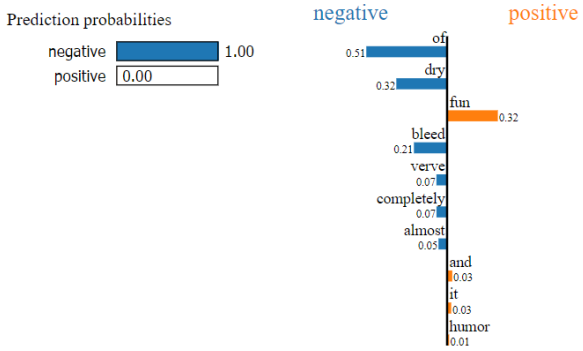

Fig. 7. Negative Review Correctly Classified by Roberta LSTM Model

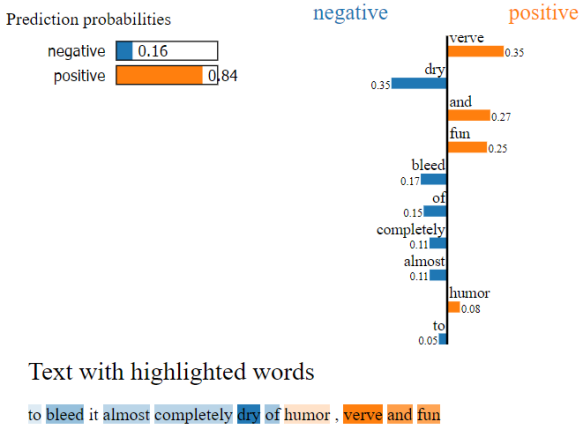

Fig. 8. Negative Review Misclassified by Bert LSTM Model

Now we will see how the two models behave while classifying a negative review "to bleed it almost completely dry

of humor, verve and fun" using Fig 7 and 8. Roberta LSTM Model correctly classifies this review as negative giving more emphasis on the words 'dry', 'of', and 'bleed' acknowledging the context of the review whereas BERT LSTM gives more weights on the words 'verve', 'and', and 'fun' giving no focus on the context.

VI. Conclusion & Future Works

In this paper, we have used Linear Regression, Multi Layer Perceptron, BERT LSTM, BERT without LSTM, RoBERTa LSTM models to classify movie review sentiments from text data. After comparing the performance of different models, we can say that RoBERTa LSTM performed better than all the other models. Our work proves the effectiveness of RoBERTa LSTM compared to other traditional machine learning approaches and BERT models. Then by utilizing LIME algorithm, we interpret the model predictions and find out the reasons behind wrong predictions made by the models. In future we plan to do multiclass classification of movie reviews including the neutral reviews of the dataset and see the inner workings of the black box models .

References

[1] Gerben Bakker. The economic history of the international film industry.

[2] Jule Selbo. *Film genre for the screenwriter*. Routledge, 2014.

[3] Sakshi Bansal, Chetna Gupta, and Anuja Arora. User tweets based genre prediction and movie recommendation using lsi and svd. In *2016 Ninth International Conference on Contemporary Computing (IC3)*, pages 1–6. IEEE, 2016.

[4] Saham Barza and Mehran Memari. Movie genre preference and culture. *Procedia-Social and Behavioral Sciences*, 98:363–368, 2014.

[5] Julia Stoll. Favorite movie genres in the u.s. by gender 2018, Jan 2021.

[6] Nilüfer Öcel. Popular cinema films in the context of turkey: Films of 2000's. *Istanbul University*, 2004.

[7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[8] Pimwadee Chaovalit and Lina Zhou. Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proceedings of the 38th annual Hawaii international conference on system sciences*, pages 112c–112c. IEEE, 2005.

[9] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50, 2006.

[10] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

[11] Atiqur Rahman and Md Sharif Hossen. Sentiment analysis on movie review data using machine learning approach. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4. IEEE, 2019.

[12] Gurshobit Singh Brar and Ankit Sharma. Sentiment analysis of movie review using supervised machine learning techniques. *International Journal of Applied Engineering Research*, 13(16):12788–12791, 2018.

[13] Kuat Yessenov and Saša Misailovic. Sentiment analysis of movie review comments. *Methodology*, 17:1–7, 2009.

[14] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[15] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] S Anbukkarasi and S Varadhaganapathy. Analyzing sentiment in tamil tweets using deep neural network. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pages 449–453. IEEE, 2020.

[18] Lysimachos Maltoudoglou, Andreas Paisios, and Harris Papadopoulos. Bert-based conformal predictor for sentiment analysis. In *Conformal and Probabilistic Prediction and Applications*, pages 269–284. PMLR, 2020.

[19] Iti Chaturvedi, Edoardo Ragusa, Paolo Gastaldo, Rodolfo Zunino, and Erik Cambria. Bayesian network based extreme learning machine for subjectivity detection. *Journal of The Franklin Institute*, 355(4):1780–1797, 2018.

[20] Kian Long Tan, Chin Poo Lee, Kalaiarasi Sonai Muthu Anbananthen, and Kian Ming Lim. Roberta-lstm: A hybrid model for sentiment analysis with transformer and recurrent neural network. *IEEE Access*, 10:21517–21525, 2022.

[21] Sajib Kumar Saha Joy, Dibyo Fabian Dofadar, Riyo Hayat Khan, Md Sabbir Ahmed, and Rafeed Rahman. A comparative study on covid-19 fake news detection using different transformer based models. *arXiv e-prints*, pages arXiv–2208, 2022.

[22] Hui Liu, Qingyu Yin, and William Yang Wang. Towards explainable nlp: A generative explanation framework for text classification. *arXiv preprint arXiv:1811.00196*, 2018.

[23] Philip H Swain and Hans Hauska. The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15(3):142–147, 1977.

[24] Dennis W Ruck, Steven K Rogers, and Matthew Kabrisky. Feature selection using a multilayer perceptron. *Journal of Neural Network Computing*, 2(2):40–48, 1990.

[25] Raymond E Wright. Logistic regression. 1995.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] P Jonathon Phillips, Carina A Hahn, Peter C Fontana, David A Broniatowski, and Mark A Przybocki. Four principles of explainable artificial intelligence. *Gaithersburg, Maryland*, 2020.