ISSN 2063-5346

PROCESS OF TEXT SENTIMENT ANALYSIS USING DEEP LEARNING FOR DOUBLE NEGATION ON PRODUCT REVIEWS

Ramandeep Sandhu[1], Abhishek Yadav[2], Kartikey Sahu[3], Nitin Jaiswal[4], Piyush Pushkar[5], Mohammad Faiz[6*]

Article History: Received: 01.02.2023 Revised: 07.03.2023 Accepted: 10.04.2023

Abstract

In the world of E-commerce and customer interest in giving their sentiments online is increasing at fast pace. Sentiments can be of three forms such as positive, negative or even sometimes sentiment in neutral form. On the other side, text-opinions which are given by customer using their own words choice is really a great challenge for reputation of a product company. In this paper, the proposed approach is working on analysis of double negation which is actually residing in negative comments but the real meaning of such comments is towards high rank of product as well as for company which is dealing with that product. A novel approach using a Long Short-Term Memory (LSTM) neural network for end-to-end sentiment analysis along with the inclusion of negation identification is implemented in this study. Our approach uses the word embedding to transform the text data into a numerical format, and introduces a Hyperparameter tuning, performed using a grid search approach to find the optimal combination of hyperparameters for the EmbedLSTM model. The approach uses a dataset obtained from Kaggle with sentiments having negation at double level. The performance of the model is evaluated using various metrics such as accuracy, precision, recall, F1 score, and specificity. Our methodology provides a rigorous approach to developing a sentiment analysis model that can be applied to a variety of datasets.

Keywords: Sentiment Analysis, Word Embedding, LSTM, Machine Learning, Natural Language Processing

[1,2,3,4,5,6*] School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India

ramandeep.28362@lpu.co.in , abs.yadav12@gmail.com , heykartikey@gmail.com , nits@duck.com , piyushpushkar1001@gmail.com, faiz.techno20@gmail.com

DOI:10.31838/ecb/2023.12.s1-B.203

1. Introduction

Now-a-days, providing feedback on a product has become a popular trend. Consumers rely on reading reviews and giving their own evaluations. Sentiment analysis of single word comments with clear meanings is easy to perform. However, text feedback can be categorized as positive, negative, or neutral. It is important to monitor both positive and negative reviews to track the product's reputation and competitors. Sentiment analysis has become an important field in natural language processing (NLP), particularly for analysing customer reviews of products and services. However, the presence of double negation in text can complicate sentiment analysis, leading to inaccurate results.

With the increasing availability of online platforms and social media, customers now have an unprecedented level of influence over businesses through their feedback and reviews. These reviews can significantly impact the purchase decisions of other customers and shape the reputation of a business in the eyes of potential customers. Sentiment analysis is a crucial subfield of natural language processing that has emerged as an essential tool for analyzing customer feedback and reviews. Its goal is to extract subjective information from text, such as opinions, emotions, and attitudes, and categorize them as positive, negative, or neutral. However, sentiment analysis becomes challenging when it comes to double negation, where the use of two negative words or phrases in a sentence can cause confusion in determining the sentiment. To overcome this challenge, deep learning techniques, such as convolutional neural networks (CNNs) and word embeddings, have shown immense potential in enhancing the accuracy of sentiment analysis. Sentiment analysis is a popular application of natural language processing that aims to extract and interpret emotions and opinions from text data. In this paper, we propose a methodology for developing an effective sentiment analysis model using a Long Short-Term Memory (LSTM) neural network. We use a dataset obtained from Kaggle and pre-process it by removing irrelevant features, handling missing values, and applying text cleaning techniques such as removing punctuation and invalid characters. We then apply word embedding to transform the text data into a numerical format.

This paper aims to explore the process of text sentiment analysis utilizing deep learning for double negation in product reviews. The primary objective is to develop an effective approach that can handle double negation and provide businesses with insightful analyses of customer sentiment, thus improving their accuracy in determining customer feedback. Sentiment analysis can be categorized as shown in **Fig. 1**.

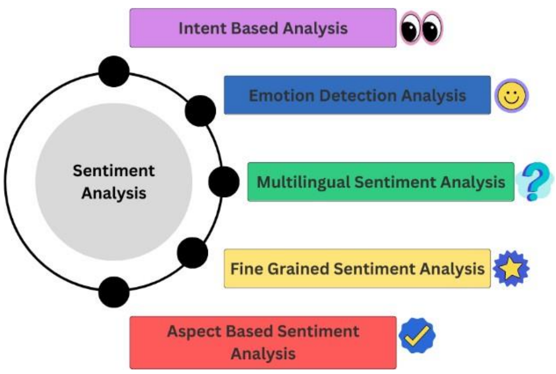

Fig 1: Various categories of Sentiment Analysis

1.1. Intent Based Analysis: It involves identifying the intended action or purpose behind a piece of text, such as a customer inquiry or support ticket. It uses natural language processing (NLP) techniques to analyse the language and structure of the text to determine the user's intent, such as making a purchase, asking a question, or expressing a concern. By understanding the intent behind customer inquiries, businesses can provide more accurate and relevant responses, improve customer satisfaction, and identify areas for process improvement.

1.2. Emotion Detection Analysis: It is a technique utilized to identify and extract emotions or moods that are expressed in each text or speech. **Fig. 2** shows various emotions shown by customers.

Fig 2: Emotions shown by customers

By comprehending the emotional state of customers, businesses can gain valuable insights into their preferences and needs and tailor their products or services accordingly to better meet their expectations.

1.3. Multilingual Sentiment Analysis: Automatically identifying and categorizing opinions expressed in text data, such as reviews, social media posts, and customer feedback, in multiple languages. The goal of this analysis is to determine the polarity of the expressed sentiment, whether it is positive, negative, or neutral, and to understand the overall sentiment trends in different languages across diverse cultures and regions. This technique is commonly used by businesses and organizations to gain insights into customer opinions and attitudes towards their products or services in global markets.

1.4. Fine Grained Sentiment Analysis: It is a more detailed and nuanced approach to sentiment analysis that goes beyond simply identifying whether a text is positive, negative, or neutral. Instead, it aims to identify the sentiment polarity towards specific aspects or attributes of a product, service, or entity mentioned in the text.

1.5. Aspect based Sentiment Analysis: The approach involves breaking down a text into its constituent parts, such as sentences or phrases, and then identifying the aspects or features mentioned in each part. Sentiment analysis is then performed on each aspect, which allows for a more detailed and nuanced understanding of the sentiment expressed in the text towards each specific feature.

2. Related Work

There have been many works done in this field by various authors. The study conducted by **Azemi et al. (2020)** [1] used semi-structured interviews and focus group interviews to gather data from 48 millennials in Albania and Kosovo about their perception of negative word-of-mouth (nWOM) on social media platforms. A thematic approach was used for data analysis. The study included 15 open-ended interview questions, and the researchers received the thread via email 24 hours before the in-depth and focus group interviews. The study also highlighted three types of online nWOM and three types of customers. The study only focused on millennials, so it is not clear how people of different ages understand nWOM. Future studies should be conducted to broaden our understanding of nWOM and its impact on different generations and settings. Furthermore, the complainant-recipient model of online nWOM should be empirically tested to ensure that the findings apply to various contexts and situations.

Dang et al. (2020) [2] conducted a study on sentiment analysis using deep learning models, including DNN, RNN, and CNN, and techniques such as word embedding and TF-IDF. They found that CNN offers the best balance between processing speed and result accuracy, while RNN has the highest accuracy when combined with word embedding but takes ten times longer than CNN. The study used datasets such as Sentiment140, Cornell Movie Reviews, Tweets SemEval, Tweets Airline, IMDB Movie Reviews, Book Reviews, and Music Reviews. The results showed that models derived from Tweets Airline dataset, which focuses on specific issues, perform better than those derived from general theme datasets. The study suggests that future research could focus on hybrid approaches to improve sentiment categorization accuracy and better balance processing speed and result accuracy.

The study by **Singh et al. (2021)** [3] compares various techniques for sentiment analysis, including Long Short-Term Memory (LSTM), Bidirectional LSTM, linear SVM, probabilistic Conditional Random Fields, Dijkstra Algorithm, RNN, and Hidden Markov Model. The study implements these techniques in TensorFlow and uses datasets from various domains such as movies, books, cars, computers, cookware, hotels, music, phones, and a Conan Doyle story. The study concludes that a deep neural network model based on LSTM is best for managing negations and nonlinear language models based on LSTM outperform more established models based on SVM, HMM, or CRF. The best outcome is achieved using Bidirectional LSTM.

In their 2021 study, **Mukherjee et al.** [4] used techniques such as Multinomial Naive Bayes (MNB), Support Vector Machines (SVM), Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), TF-IDF feature extraction, and SentiWordNet to analyse product reviews. They specifically considered negations in the reviews and handled double negationsusing the '_NEG' suffix. They compared the performance of classifiers using various parameters and found that the sentiment classifier performed better when conventional text classification classifiers were combined with negationidentification.

In 2017, **Mukherjee et al.** [5] proposed the negation parser, NegAIT, to identify morphological, syntactical, and double negations from medical corpora. They classify “easy” and “difficult” texts with extremely high accuracy and precision using distinct types of negations as the predictors.

3. Methodology

In this paper, we propose a process for text sentiment analysis using deep learning for handling double negation on product reviews. The proposed method involves importing a textual dataset, applying pre-processing techniques and selecting features such as customer id, timestamp, and product review submitted. Unique product reviews are compiled, and values below satisfactory levels are extracted[6-9]. Negative grammar word datasets are imported for transformation using deep learning. If a review statement contains two consecutive words from the negative grammar word dataset, it is considered a positive review and assigned a rank value towards positive product reputation. Once the network is trained, it could be used to automatically analyse new product reviews, providing insights into customer sentiment and identifying areas where improvements could be made. The proposed methodology consists of several steps to develop an effective sentiment analysis model. First, the dataset obtained from Kaggle is pre-processed by removing irrelevant features, handling missing values, and applying text cleaning techniques such as removing punctuation and invalid characters. Next, word embedding is applied to transform the text data into a numerical format. Fig 1 shows the overall architecture of the proposed model.

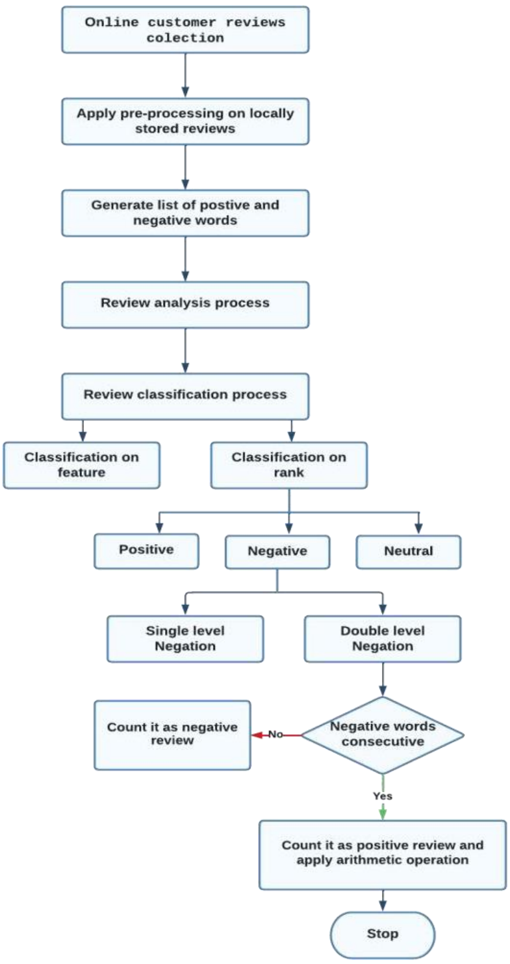

Fig 1. Proposed deep learning based methodology

Hyperparameter tuning is then performed using a grid search approach to find the optimal combination of hyperparameters for the EmbedLSTM Model. The hyperparameters considered include the number of LSTM units, dropout rate, recurrent dropout rate, optimizer, number of epochs, and batch size[10][11].

The EmbedLSTM Model architecture is then defined based on the optimal hyperparameters. The model includes an embedding layer, LSTM layer, dense layers with ReLU activation function and Sigmoid activation function in the last layer for binary classification.

Finally, the performance of the model is evaluated using various metrics such as accuracy, precision, recall, F1 score, and specificity. This allows for a comprehensive understanding of the model's ability to correctly classify sentiment. Overall, this methodology provides a rigorous approach to developing a sentiment analysis model that can be applied to a variety of datasets[12][13].

3.1. Data Gathering

The dataset used in this study was obtained from Kaggle, a platform for data science competitions and datasets. The dataset was downloaded as a CSV file and uploaded to Google Drive[14]. The dataset was then accessed and processed using Google Colaboratory, a cloud-based platform for running Jupyter notebooks. The dataset consists of 189911 samples, with each sample representing Product Name, Review, Rating, and other but not relevant information. The dataset was cleaned before feeding it to our proposed model as it had invalid rating value, and invalid characters in review. We added 55 instances of double negated reviews in the dataset by ourselves to check if our model could classify it correctly which is the chief purpose of our study. The distribution of positive and negative instances (after converting rating to sentiment) are as given in **Table 1**.

Table 1: Class Distribution in the Dataset

Sr. No.	Class	Count
1	Positive Instances	1,64,070
2	Negative Instances	25,841
3	Total Instances	1,89,911

3.2. Programming Environments

In Deep learning, Python is the most popular programming language. It offers a wide range of libraries that may be used to put different deep-learning methods into practice [15][16]. Because of its versatility, abundance of open-source libraries, platform freedom, and ease of use, Python was our choice for this study.

3.3. Data Pre-processing

Data pre-processing is a crucial step in sentiment analysis, as it helps to prepare the raw text data before it is fed into a machine learning model for analysis. Raw text data may contain several types of noise and inconsistencies, such as special characters, punctuation, capitalization, and misspellings, which can negatively impact the performance of the machine learning model.

By applying various data pre-processing techniques such as removing stop words, stemming, lemmatization, and normalisation, we can remove the noise and inconsistencies in the data and transform it into a format that is easier for the machine learning model to interpret. This can improve the accuracy and reliability of the sentiment analysis results [17][18].

Therefore, in this section, we performed several pre-processing steps on the dataset to ensure the quality and suitability of the data for analysis. The following subsections outline the specific pre-processing steps that were applied.

3.3.1. Data Cleaning

The dataset was first examined for missing values and irrelevant features. We removed the 'ProductName' and 'Price' columns as they were not relevant to our analysis. We also removed any rows that did not have a 'Summary' value. Finally, we removed any rows where the 'Rate' value was not within the range of 1 to 5.

3.3.2. Text Cleaning

After cleaning the data, we performed text cleaning to ensure consistency and standardisation of the 'Summary' values. We first removed all punctuation marks from the 'Summary' values. Then, we removed any invalid characters that could cause errors during analysis. This was achieved using regular expressions to remove non-alphanumeric characters [19].

3.3.3. Sentiment Labelling

To perform sentiment analysis, we needed to label the reviews as either positive or negative [20]. We used the 'Rate' column for this purpose, where ratings of 1 and 2 were labelled as negative (0) and ratings of 3 and above were labelled as positive (1) [20][21].

3.3.4. Data Reduction

Thereafter, we removed the 'Rate' column from the dataset, as it was no longer needed for our analysis [22][23].

3.3.5. Numeric Vector

Finally, Word embedding is applied to transform the text data, the reviews, into a dense vector of numbers. The resulting dataset was then used for further analysis [24][25].

3.4. Building the EmbedLSTM Model

This model is a sequential neural network with multiple layers used for sentiment analysis. It uses an embedding layer to transform the input text into dense vectors, followed by a Long Short-Term Memory (LSTM) layer to capture temporal dependencies in the input sequence. The LSTM layer is followed by a flatten layer to reduce the dimensionality of the output, and then two dense layers with ReLU activation functions. The final dense layer uses a sigmoid activation function to output a probability value between 0 and 1, which represents the predicted sentiment of the input text. The overview of the architecture of the model is given in **Fig 2.** [26][27].

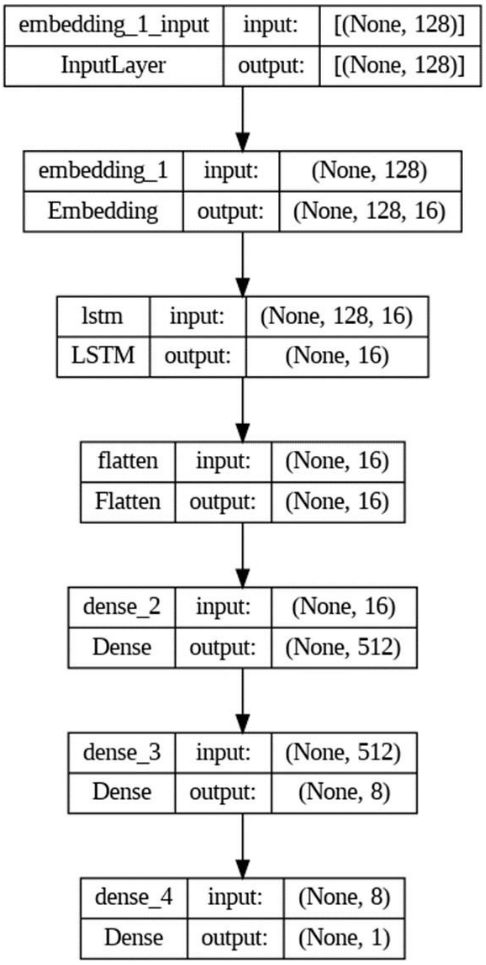

Fig 2: Architecture of EmbedLSTM Model

3.5. Hyperparameter Tuning

Hyperparameter tuning is an essential step in building deep learning models. The performance of a deep learning model is highly dependent on the values of its hyperparameters. Therefore, it is essential to tune the hyperparameters to achieve optimal performance. In this study, we used the GridSearchCV module of scikit-learn to search for the best combination of hyperparameters for our EmbedLSTM Model [28][29].

We defined the following hyperparameters for our model: the number of LSTM units, dropout rate, recurrent dropout rate, optimizer, number of epochs, and batch size. We defined a range of values for each hyperparameter to be searched during the grid search. The number of LSTM units was set to 32, 64, and 128. The dropout rate was set to 0.2, 0.4, and 0.6. The recurrent dropout rate was set to 0.2, 0.4, 0.6, and 0.8. The optimizer was set to 'adam' and 'sgd'. The number of epochs was set to 10, 20, and 30. The batch size was set to 16, 32, and 64.

We used the KerasClassifier module to create an instance of our EmbedLSTM Model that can be used by GridSearchCV. We also defined the scoring parameter for GridSearchCV to include F1-score, precision, recall, and specificity. The F1-score is a harmonic mean of precision and recall, which is a suitable metric for imbalanced datasets. Precision is the number of true positive predictions divided by the sum of true positive and false positive predictions. Recall is the number of true positive predictions divided by the sum of true positive and false negative predictions. Specificity is the number of true negative predictions divided by the sum of true negative and false positive predictions.

The GridSearchCV module searches for the best combination of hyperparameters using cross-validation. We set the cv parameter to 3, which means that the dataset was split into three folds for cross-validation. The refit parameter was set to 'f1_score', which means that the best model was selected based on the F1-score. After the grid search, the best hyperparameters were used to train the EmbedLSTM Model on the entire dataset [30].

3.6. Model Evaluation

The evaluation of the model is essential to understand its performance and effectiveness in predicting the sentiment of the reviews accurately. To evaluate the performance of the model, we used various metrics, including F1 score, accuracy, precision, recall, and specificity. Each metric provides unique insights into the model's performance and helps in determining its effectiveness in classifying the reviews into positive or negative sentiment. In the following subsections, we will describe each metric in detail, including its formula and interpretation.

3.6.1. F1 Score

A machine learning evaluation metriccalled the F1 score assesses a model's accuracy. A weighted harmonic mean of accuracy and sensitivity is the F1 score. It is a useful tool that strikes a balance between sensitivity and precision or accuracy. It is calculated using the formula:

$$\mathrm{F1-Score} = \frac{2 * \mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

where precision is the ratio of true positive (TP) predictions to the total number of positive predictions, and recall is the ratio of TP predictions to the total number of actual positive cases.

3.6.2. Accuracy

Accuracy is a metric that measures the proportion of correct predictions made by the model over the total number of predictions. Accuracy reveals how frequently the ML model was overall correct. It is calculated using the formula:

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

where TP means the number of true positive predictions, TN means the number of true negative predictions, FP means the number of false positive predictions, and FN means the number of false negative predictions.

3.6.3. Precision

Precision measures the proportion of true positive predictions to the total number of positive predictions made by the model. It is inversely related to sensitivity, which means that as precision rises, sensitivity falls. It is calculated using the formula:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

where TP means the number of true positive predictions and FP means the number of false positive predictions.

3.6.4. Recall

Recall measures the proportion of true positive predictions to the total number of actual positive cases in the dataset. It i1s calculated using the formula:

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

where TP means the number of true positive predictions and FN means the number of false negative predictions.

3.6.5. Specificity

Specificity measures the proportion of true negative predictions to the total number of actual negative cases in the dataset. It assesses the percentage of genuine negatives that the model accurately detects. It is calculated using the formula:

2

$$\mathrm{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

4. Result

We trained our proposed model, EmbedLSTM, on a dataset consisting of 1,89,911 instances including 55 manually added double negated reviews. The dataset was pre-processed as per the steps described in **Section 3.3**.

We used the following hyperparameters for the model:

- vocab_size = 1000
- embed_dim = 16
- max_len = 128

The vocab_size parameter determines the size of the vocabulary used by the model, with a maximum of 1000 unique words. The embed_dim parameter specifies the dimensionality of the word embeddings used by the model, with each word represented by a vector of 16 dimensions. The max_len parameter specifies the maximum length of the input text, with a limit of 128 tokens.

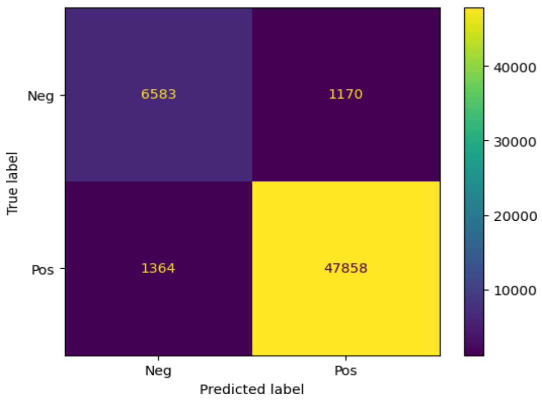

Fig 4: Confusion Matrix for EmbedLSTM Model

In the **Fig 4**, confusion matrix, the true values (actual sentiment of the text) are represented on the y-axis, while the predicted values (sentiment predicted bythe model) are represented on the x-axis. The matrix contains four cells, each representing the number of instances where the model correctly or incorrectly predicted the sentiment of the text.

The top-left cell (6583) represents the number of instances where the model correctly predicted a negative sentiment. This means that the text was negative, and the model predicted it correctly.

The top-right cell (1170) represents the number of instances where the model predicted a positive sentiment incorrectly. This means that the text was negative, but the model incorrectly predicted it to be positive.

The bottom-left cell (1364) represents the number of instances where the model predicted a negative sentiment incorrectly. This means that the text was positive, but the model incorrectly predicted it to be negative.

The bottom-right cell (47858) represents the number of instances where the model correctly predicted a positive sentiment. This means that the text was positive, and the model predicted it correctly.

The model was trained for 30 epochs and achieved high performance on the dataset, with an accuracy of 95.55%, precision of 97.61%, an F1 score of 97.42%, and a recall/sensitivity score of 97.23%. These results indicate that our model can accurately classify instances of positive, and negative sentiment.

We also calculated the specificity score of our model, which measures its ability to correctly identify negative instances. Our model achieved a specificity score of 84.91%, indicating that it correctly identified a large majority of negative instances in the dataset.

Table 2: Performance Metrics for our proposed model

Metric	Score
Accuracy	95.55%
Precision	97.61%
F1 Score	97.42%
Sensitivity	97.23%
Specificity	84.91%

Overall, these results demonstrate the effectiveness of our sentiment analysis model on the given dataset. However, it is important to note that the performance of the model may vary depending on the characteristics of the dataset and the specific application. Further experimentation and evaluation may be necessary to determine the generalizability of our model to other datasets and contexts.

5. Conclusion

In this work, deep learning approaches with architecture and parameter tuning, the EmbedLSTM Model is proposed. The primary objective of this study was to analyse the sentiment of product and customer reviews and effect of double negation reviews present in the dataset. The work focused on sentiment analysis of text-based product reviews, which are generated from the feedback of millions of customers on several e-commerce platforms using deep learning techniques such as LSTM along with different techniques like Word Embedding and GridSearchCV for hyper tuning the model. The concerned text in proposed technique is working on negation at double level which exists in negative revies. The combination of these different techniques with EmbedLSTM Model has significantly improved our results. Upon evaluation using various metrics like accuracy, precision, etc., the model performed well and had better results predicted. In the age of data, most businesses make significant investments in comprehending data and deriving insights from it to comprehend customer input that is acquired from a variety of sources, including social media platforms like Amazon, Facebook, Twitter, and others. The significance of the embedding layer in sentiment classification has been proven.

References

1. P. Mukherjee, Y. Badr, S. Doppalapudi, S. M. Srinivasan, R. S. Sangwan, and R. Sharma, “Effect of negation in sentences on sentiment analysis and polarity detection,” *Procedia Comput. Sci.*, vol. 185, pp. 370–379, 2021.
2. Y. Azemi, W. Ozuem, K.E. Howell, “The effects of online negative word-of-mouth on dissatisfied customers: A frustration–aggression perspective”, Psychology & Marketing, vol. 37, 2020, pp. 564-577 <DOI: 10.1002/mar.21326>.
3. M. Hina, M. Ali, A. R. Javed, F. Ghabban, L. A. Khan, and Z. Jalil, “Sefaced: Semantic-based forensic analysis and classification of e-mail data using deep learning,” *IEEE Access*, vol. 9, pp. 98398–98411, 2021.
4. https://shodhganga.inflibnet.ac.in/handle/10603/377504
5. N.C. Dang, M.N. Moreno-García, F. De la Prieta, “Sentiment analysis based on deep learning: A comparative study”, Electronics, vol. 9(3), 2020, pp. 483-512 <DOI: 10.3390/electronics9030483>.
6. Sandhu R., Applying Opinion Mining to Organize WebOpinions. International Journal of Computer Science, Engineering and Applications International Journal of Computer Science, Engineering and Applications, Volume 1, pp 82-89; https://doi.org/10.5121/ijcsea.2011.1408, IJCSEA
7. P.K. Singh, S. Paul, “Deep learning approach for negation handling in sentiment analysis”, IEEE Access, vol. 9, 2021, pp. 102579-102592 < DOI: 10.1109/ACCESS.2021.3095412>.
8. Sandhu R., Lakhwani K. 2021. Improved Scientific Workflow Scheduling Algorithm with distributed HEFT Ranking and TBW

Scheduling Method in International Conference on Wireless Sensor Networks, Ubiquitous Computing and Applications. ISSN- 2367-3370. Vol no. 244. https://doi.org/10.1007/978-981-16-2919-8.

9. P. Mukherjee, Y. Badr, S. Doppalapudi, S.M. Srinivasan, R.S. Sangwan, R. Sharma, "Effect of negation in sentences on sentiment analysis and polarity detection", Procedia Computer Science, vol. 185,2021, pp. 370-379 <DOI: 10.1016/j.procs.2021.05.038>.
10. Sandhu R. A Novel method to find score value for Online Opinions. (International Journal of Computational Science and Information Technology), ISSN: 2320-7442 (online): 2320-8457 (Print), February 2013, Volume 1, Number 1. IJCSITY
11. Choudhary, S., Narayan, V., Faiz, M., & Pramanik, S. (2022). Fuzzy approach-based stable energy-efficient AODV routing protocol in mobile ad hoc networks. In Software Defined Networking for Ad Hoc Networks (pp. 125-139). Cham: Springer International Publishing.
12. A. Iqbal, R. Amin, J. Iqbal, R. Alroobaea, A. Binmahfoudh, and M. Hussain, "Sentiment Analysis of Consumer Reviews Using Deep Learning," *Sustainability*, vol. 14, no. 17, p. 10844, 2022.
13. Mukherjee, Partha, Gondy Leroy, David Kauchak. (2017) "NegAIT: A new parser for medical text simplification using morphological, sentential and double negation." Journal of biomedical informatics, 69: 55-62.
14. N. Ali Hakami and H. A. Hosni Mahmoud, "Deep Learning Analysis for Reviews in Arabic E-Commerce Sites to Detect Consumer Behavior towards Sustainability," *Sustainability*, vol. 14, no. 19, p. 12860, 2022.
15. M. Y. A. Salmony and A. R. Faridi, "S[1] M. Y. A. Salmony and A. R. Faridi, 'Supervised Sentiment Analysis on Amazon Product Reviews: A survey,' in 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), 2021, pp. 132–138.upervised Sentiment Analysis on Amazon P," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, 2021, pp. 132–138.
16. Mall, P. K., Narayan, V., Pramanik, S., Srivastava, S., Faiz, M., Sriramulu, S., & Kumar, M. N. (2023). FuzzyNet-Based Modelling Smart Traffic System in Smart Cities Using Deep Learning Models. In S. Pramanik & K. Sagayam (Eds.), Handbook of Research on Data-Driven Mathematical Modeling in Smart Cities (pp. 76-95). IGI Global. https://doi.org/10.4018/978-1-6684-6408-3.ch005
17. Faiz, M., Daniel, A.K. (2022). Wireless Sensor Network Based Distribution and Prediction of Water Consumption in Residential Houses Using ANN. In: Misra, R., Kesswani, N., Rajarajan, M., Veeravalli, B., Patel, A. (eds) Internet of Things and Connected Technologies. ICIoTCT 2021. Lecture Notes in Networks and Systems, vol 340. Springer, Cham. https://doi.org/10.1007/978-3-030-94507-7_11
18. Faiz, M., & Daniel, A. K. (2022). Threats and challenges for security measures on the internet of things. Law, State and Telecommunications Review, 14(1), 71-97.
19. Faiz, M., Daniel, A.K. A multi-criteria cloud selection model based on fuzzy logic technique for QoS. Int J Syst Assur Eng Manag (2022). https://doi.org/10.1007/s13198-022-01723-0
20. Faiz, M., & Shanker, U. (2016, March). Data synchronization in distributed client-server applications. In 2016 IEEE International Conference on Engineering and Technology (ICETECH) (pp. 611-616). IEEE.
21. Babu, S. Z., et al. "Abridgement of Business Data Drilling with the Natural Selection and Recasting Breakthrough: Drill Data With GA." Authors Profile Tarun Danti Dey is doing Bachelor in LAW from Chittagong Independent University, Bangladesh. Her research discipline is business intelligence, LAW, and Computational thinking. She has done 3 (2020).
22. Faiz, Mohammad, et al. "IMPROVED HOMOMORPHIC ENCRYPTION FOR SECURITY IN CLOUD USING PARTICLE SWARM OPTIMIZATION." Journal of Pharmaceutical Negative Results (2022): 4761-4771.
23. Narayan, Vipul, A. K. Daniel, and Pooja Chaturvedi. "E-FEERP: Enhanced Fuzzy

based Energy Efficient Routing Protocol for Wireless Sensor Network." Wireless Personal Communications (2023): 1-28.

24. Paricherla, Mutyalaiah, et al. "Towards Development of Machine Learning Framework for Enhancing Security in Internet of Things." Security and Communication Networks 2022 (2022).

25. Tyagi, Lalit Kumar, et al. "Energy Efficient Routing Protocol Using Next Cluster Head Selection Process In Two-Level Hierarchy For Wireless Sensor Network." Journal of Pharmaceutical Negative Results (2023): 665-676.

26. Sawhney, Rahul, et al. "A comparative assessment of artificial intelligence models used for early prediction and evaluation of chronic kidney disease." Decision Analytics Journal 6 (2023): 100169.

27. Srivastava, Swapnita, et al. "An Ensemble Learning Approach For Chronic Kidney Disease Classification." Journal of Pharmaceutical Negative Results (2022): 2401-2409.

28. Mall, Pawan Kumar, et al. "Early Warning Signs Of Parkinson's Disease Prediction Using Machine Learning Technique." Journal of Pharmaceutical Negative Results (2022): 4784-4792.

29. Mall, Pawan Kumar, et al. "FuzzyNet-Based Modelling Smart Traffic System in Smart Cities Using Deep Learning Models." Handbook of Research on Data-Driven Mathematical Modeling in Smart Cities. IGI Global, 2023. 76-95.

30. Narayan, Vipul, et al. "Deep Learning Approaches for Human Gait Recognition: A Review." 2023 International Conference on Artificial Intelligence and Smart Communication (AISC). IEEE, 2023