Hate and Sarcastic Writing Detection

M. Musab Rasheed, Salman Younus, Ali Faheem
SBASSE

Natural Language Processing | Semester Project
Intstructor: Dr. Asim Karim | 30th April, 2021

Abstract

Today we are living in a digital world and the Internet has revolutionized the way in which people communicate and share information with each other; social media / online networking are the true examples. People express their thoughts using different social media platforms provided like Facebook, Instagram, Twitter etc. and expressing their views about any ongoing trend in a positive or negative way making web based social networking imperative in our society. Besides there are good things about online social networking, there are some serious challenges/issues and hatred contents on the internet is one of them. However, the efficiency and accuracy of sentiment analysis is being hindered by the challenges encountered in natural language processing (NLP). In recent years, it has been demonstrated that deep learning models are the promising solution to the challenges of NLP. This report reviews the latest studies that have employed deep learning to solve sentiment analysis problems. Models using LSTM and word embedding have been applied to the dataset. Finally, a comparative study has been conducted on the IMDB Dataset using different models.

1 Introduction

The Internet and technologies have changed the perspective of today's living known as the digital world. In which everything is connected and communicating without any hindrance. People are communicating over the internet using social platforms and expressing their thoughts by sharing the blogs and articles, tweeting about any topics and expressing over posts. The technologies are adapting towards optimization over the huge amount of data. As these technologies grow exponentially the problems are arriving alongside. One of the major problems is text classification for optimizing the search results. To address this problem there are multiple ways (i.e., sentiment analysis) to do that like machine learning and fuzzy rule systems.

Sentiment analysis is one of the most interesting topics in Natural Language processing. It is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes [1]. So sentiment analysis is widely used in recommendation systems i.e., Recommendation of any product, services, posts, blogs, online courses movies etc.. So based on comments, sentiments or reviews a model can predict or recommend that specific kind. Which is helpful for others to think about the product, services or anything using that predicted information.

It had always been an important piece of information during the decision process [2], like recommendation of clothes in Amazon based on the sentiment analysis of user reviews. Applications of sentiment analysis have been a part of many different areas from past decades. In social media such as Facebook and Snapchat, there are tremendous stories and data which can be used for advertising, marketing, recommending new friend relationship, etc. But social posts and tweets are full of complex lexicons, acronyms, and emoticons, which are non-trivial problems for the feature space in machine learning models. This is not only limited to social post but also applied to business intelligence models [3]. A remarkable increase in Web tools i.e., online social media and e-commerce websites where users freely express their ideas and thoughts/ emotions. Owing to this increase, huge amounts of data are generated. Therefore, sentiment analysis was introduced as a tool for automatic extraction of insight and useful information from the user-generated data. It has attracted a large number of researchers and industry communities because of its usefulness and challenges [4].

Now a days classification take a peak as there are many innovation has taken place in deep learning and classical artificial models. Many work has been done on optimization i.e., Genetic Algorithms, Particle Swarm Intelligence, Ant Colony Optimization, Artificial Bee Colony. These are heuristic in nature and select the optimum solutions from the population. Classifying document/text into respective classes using genetic algorithm [5]. In text classification, feature selection is essential to improve the classification effectiveness without it many NLP task aren't possible as they are now. Bidi Noria provides an empirical study of a feature selection method based on genetic algorithms for different text representation methods [6].

Some sentiment analysis tasks do a binary classification like positive and negative, and others may do multi-level classification like hatred, funny, neutral, sarcastic, positive etc. Twitter, Facebook, Movies Reviews are one of them. In this study we are focusing to implement different neural network models discussed in recent research work and perform evaluation.

2 Literature Review

For many years this field of sentiment analysis is the interesting research areas of scientists and researchers. In this section we will be providing a comprehensive overview of the recently proposed algorithms, enhancements, and applications in the area of sentiment analysis e.g., transfer learning, emotion detection, and building resources. Medhat, W., Hassan, A. & Korashy tried to give a full image of the sentiment analysis techniques and related fields with brief details [7]. The machine learning model approach is aligned with these studies. However, some are employing Genetic Algorithm (GA) based optimized feature selection for training ML algorithms [8]. Sentiment analysis is done on many datasets with the passage of time some famous datasets are IMDB, Twitter, Rental Prices, business loan and transactions etc. Models using term frequency-inverse document frequency (TF-IDF) and word embedding have been applied to a series of datasets [9]. Results on various classification and sequence labelling showed that the modified Bidirectional LSTM model has strong representation power, giving highly competitive performances compared to stacked BiLSTM models with similar parameter numbers [10]. In recent years attention networks have gained popularity for sentiment classification tasks. [11] proposed a hierarchical attention network for document level sentiment rating prediction of reviews. He used word level and sentence level attention mechanisms, which allow the model to pay more or less attention to individual words or sentences in constructing the representation of a document. [12] proposed a hierarchical interactive attention-based model to learn aspect-aware document representation. [13] proposed Attention-based Bidirectional CNN-RNN Deep Model (ABCDM). It utilizes two LSTM and GRU layers to extract past and future context. In contrast to simple LSTM and GRU, Attention based mechanisms was applied to put more or less emphasis on different words in a sentence. This model achieved state of art accuracy on both long and short tweets.

Many work has been done on optimisation of sentiment analysis to enhance the sentiment classification accuracy and to reduce the computational cost [14]. In the present study, Salur, M. U. & Aydin has proposed a new novel hybrid deep learning model that strategically combines different word embedding (Word2Vec, FastText, character-level embedding) with different deep learning methods (LSTM, GRU, BiLSTM, CNN). The proposed model extracts features of different deep learning methods of word embedding, combines these features and classifies texts in terms of sentiment [15]. As we observed that most of the related work employed independent techniques for sentiment analysis while using few evaluation metrics. Furthermore, they do not provide the user with the freedom to choose different algorithms, classifiers, and optimizations according to customized needs. In our study, we will give a comparative study on sentiment analysis using neural networks based on LSTM, Bidirectional LSTM, Convolutional Neural Networks (CNN) and BERT model and will also evaluate these models to check the accuracy on IMDB Dataset to solve sentiment classification.

3 Methodology

Our approach is based on text processing methodology where we will prepare the data, apply text classification techniques, and will process and evaluate the results. The high level understanding is provided in the following figure 1.

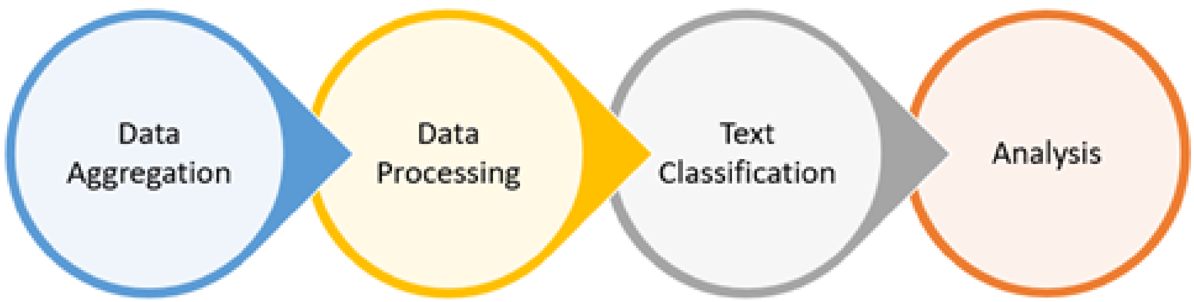


Fig. 1. Sentiment Classification Process Flow

As part of data aggregation and processing, we have done the data cleaning and data preprocessing activities. A brief summary of the steps is as follow:

3.1 Data Cleaning

Data cleaning is the part of pre-processing because data is available in raw format. To make it understandable for the model to train better we need to do the data cleaning. So, in this phase, we extracted data from the files and saved them in the memory for cleaning purposes. This stage consists of following steps.

1. Cleaning the review text, @, #, and the links from the sentences.
2. Converting the text to lower case.
3. Cleaning all the non-letter characters, including numbers.
4. Re.moving English stop words and punctuation
5. Decoding HTML to general text.
6. Stemming or lemmatization

3.1.1 Contraction Removal

Contraction removal is the pre-processing part in which we transform the word contraction to readable words. because of model limitation on text classification. For example isn't is the contraction of 'is not', so we transform them to the words.

3.1.2 Garbage Removal

In this step, unwanted characters (non-ASCII characters) including URLs, web addresses, and online links are removed from the text using customized regular expressions. These unwanted characters and words reduce the model accuracy on particular dataset. Non alphanumeric characters, punctuation are removed using regular expression.

3.1.3 Stopword Removal

Stopword removal removes very common words of a language e.g., "an", "about", "above" etc. These words usually have no impact in text classification between positive sentences or negative ones or even for multi class classification problems. We used NLTK library which contains the stopword list for finding any stopword in the data.

3.2 Sentence Pre-processing

This module includes sentence pre-processing i.e., tokenization, word stemming and lemmatization. After using all processes, the text now is in tokenized word form which is helpful during the training process.

3.2.1 Tokenization

Tokenization is the process of breaking a series of text (sentence) into words, phrases, symbols, or other meaningful elements called tokens. In order to tokenize the text, Nltk library lets you do that using WordTokenizer, which is used for these experiments. Initially, we used StringTokenizer but due to the inherent limitations of this tokenizer, we opted for much better WordTokenizer. The key point is that custom data structures are designed to hold tokens (features) and sentences (list of features) of each document.

3.2.2 Stemming and Lemmatization

Stemming and Lemmatization are the process of reducing inflected word to its base or root word (the stem). The framework uses StemPorter to convert each token to its stem form and store it in the Keyword object. Stemming is faster than lemmatization because lemmatization uses dictionary data structure and uses memory to do stemming to yield better results.

3.3 Model Engineering

In this section, We'll discuss the Neural Networks that is used to solve classification problem.

3.3.1 Neural Networks

Neural networks, also known as artificial neural networks (ANNs) are the subset of machine learning and the base of deep learning. Neural networks are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Artificial neural networks (ANNs) consist of three type of layers, an input layer, one or more hidden layers, and an output layer. Each node/neuron is connected to another and has an associated weight or threshold. If the output of any individual node is above the specified threshold value, that neuron is activated, sending layer data to the next layer of the network. Otherwise, these neurons remain inactive.

The Network tends to learn and improve the accuracy over training data input. This concept behind how these networks work in general, Let's take an example. The problem is to classify the given text, for human it's easy to perceive and figure out, but in case of ann, it involves mathematics and several layers of functions to process the text into data points and information that computer can use. The neural network can start to identify the patterns (features) that exist across the data-set and after processing, it classifies text by its similarities.

$$\begin{aligned} z &= \sum_{i=0}^{n} w_i x_i + b \\ &= w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b \end{aligned} \tag{1}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

The condition $\sum_{i=0}^{n} w_i x_i$ is the sum of the dot product of the weights and inputs of the neurons. Weights and inputs of the neurons are just the vectors and bias is the measure (or threshold).

Motivation behind it is that, **Deep Learning** has performed well in numerous fields of computer science i.e., Computer Vision, Natural Language Processing, Medical Sciences etc. These models were used to solve complex problems of real life. Text Classification is one of them. So, we are using the different neural architecture for sentiment analysis activity.

LSTM LSTM [Long short-term memory] is an artificial recurrent neural network (RNN) architecture used in deep learning. Unlike standard feed-forward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or sequence of text). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition.

Bidirectional LSTM Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems. In problems where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence.

Convolutional Neural Network CNN is the basic multi-layered neural networks used for different tasks. The architecture diagram is in the following Figure 3.

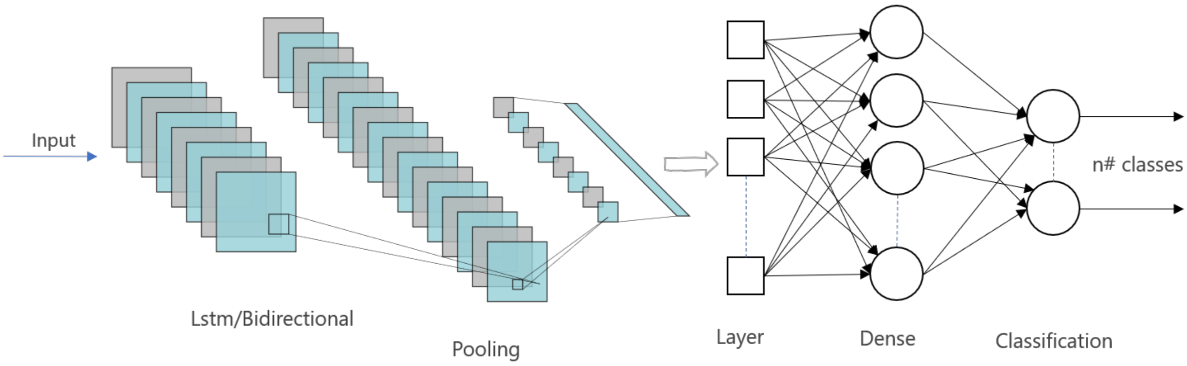


Fig. 2. Neural Network Architecture

BERT is a pre-trained unsupervised natural language processing model. BERT is deeply bi-directional, meaning it looks at the words before and after entities and context pre-trained on Wikipedia to provide a richer understanding of language. BERT is an open source machine learning framework for natural language processing (NLP). In addition to the neural networks, we have tested BERT model that is a pre-trained unsupervised natural language processing model. BERT is deeply bi-directional, meaning it looks at the words before and after entities and context pre-trained on Wikipedia to provide a richer understanding of language. BERT is an open source machine learning framework for natural language processing.

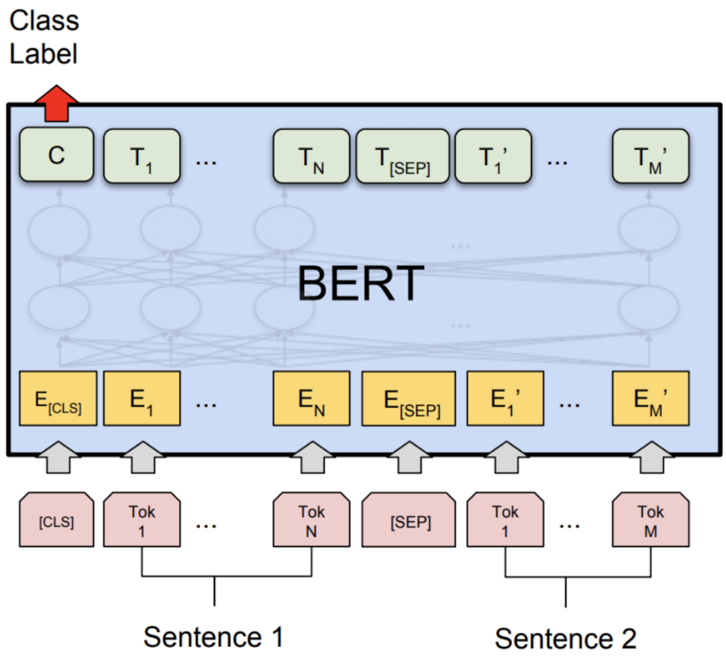


Fig. 3. Neural Network Architecture (Image Taken form [16])

It is designed to help computers understand the meaning of in in format text by using surrounding text to establish contex so called of the findings is accurate mentioned in order to address the sentiment analysis problem for IMDB dataset.

3.3.2 Classical Machine Learning Models

Machine learning is the breakthrough in the field of computer science because of the diversity. As it provides solution to most of the problems where there are numbers, data and analysis is required. But it has faced many problem as well i.e., with natural language is one major hurdle is its algorithms usually deal with numbers, and natural language is in text format. So we need to transform that text into numbers so called text vectorization. It's a fundamental step in the process of machine learning for analyzing data, and different vectorization algorithms will drastically affect end results, so we need to choose one that will deliver the results for problem like sentiment classification.

Once we are done transforming words into numbers, it will be understandable to machines and algorithms. The TF-IDF score can be fed to algorithms such as Naive Bayes and Support Vector Machines.

3.3.3 TF-IDF Vectorizer

Why does this work? Simply put, a word vector represents a document as a list of numbers, with one for each possible word of the corpus. Vectorizing a document is taking the text and creating one of these vectors, and the numbers of the vectors somehow represent the **content of the text**. TF-IDF enables us to gives us a way to associate each word in a document with a number that represents how relevant each word is in that document. Then, documents with similar, relevant words will have similar vectors, which is what we are looking for in a machine learning algorithm. So, we used it for IMDB Dataset for classifying each review/text to positive(Non Hatred) and Negative (Hatred) classes.

3.3.4 Logistic Regression

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability. It's a discriminative model and for that we use **Sigmoid function** as cost function or also known as the "logistic function" instead of a linear function.

$$Sigmoid\sigma(z) = \frac{1}{1+e^{-z}}$$

We expect our LR classifier to give us a set of outputs/classes (positive or negative) based on probability when we pass the inputs through a prediction function and returns a probability score between 0 and 1.

3.3.5 Random Forest

Random forest is a supervised learning algorithm which is used for both classification as well as regression. A forest is made up of trees, similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and selects the best solution. The idea is that a combination of learning models increases the overall result.

3.3.6 Naive Bayes

NB is the binary classifier and it is based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$. We used it for Sentiment Analysis (IMDB).

3.3.7 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression problems. But mainly used for classification problems.

In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

All the results have been shown in the section 5 and table 2. We have used the Twitter data-set for the evaluation.

4 Additional Work

4.1 Web-based Application Using Bert

The Peltarion Platform is a cloud-based operational AI platform that allows you to build and deploy your deep learning models. The service offers an end-to-end platform that lets you do everything from pre-processing your data to building models and putting them into production.

As an addition to the project, we made an application named **'Sentiment Classifier'** using BERT Model on IMDB dataset. This classify each review between Positive and Negative. So first we made the BERT model as you can see in the figure 4 So features will be passed as

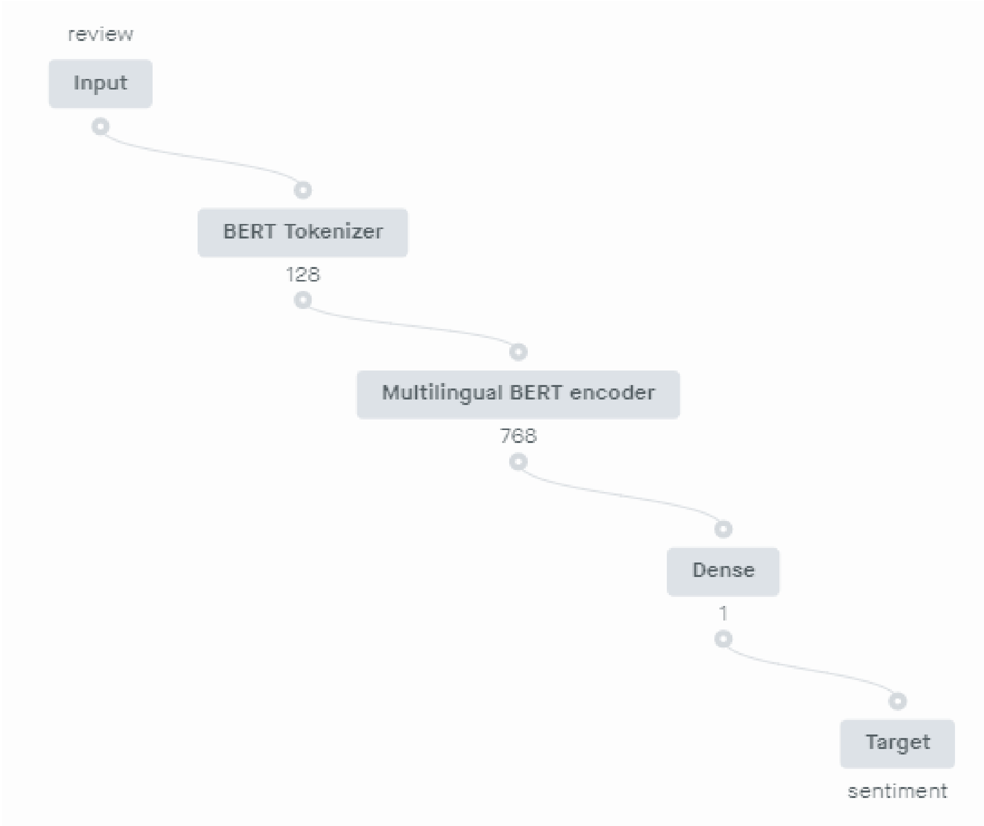
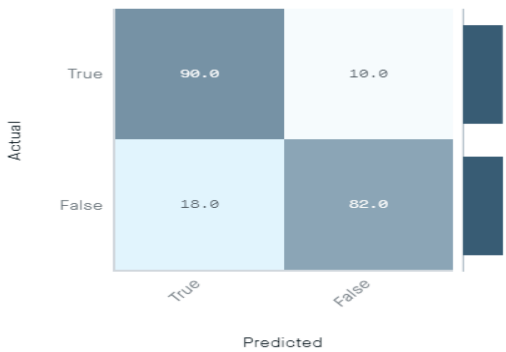


Fig. 4. BERT Model

an input to this model then tokenize each sentence and after that we used the **Multilingual BERT Version** which is different from earlier mentioned.

Metric (y axis)	V	T
Binary crossentropy (loss)	0.311	0.271
Binary accuracy	0.862	0.885
Precision	0.834	0.883
Recall	0.903	0.887
F1-score	0.867	0.885
ROC AUC	0.944	0.955
PR AUC	0.943	0.955
Binary error	0.138	0.115
Learning rate		0.000

(a) Accuracy



(b) Confusion Matrix

Fig. 5. Multilingual BERT Model on IMDB Dataset

Visulization Finaly, after training of the model we deployed it on cloud using PP' API and evaluated the Twitter texts. You can see in figure 6

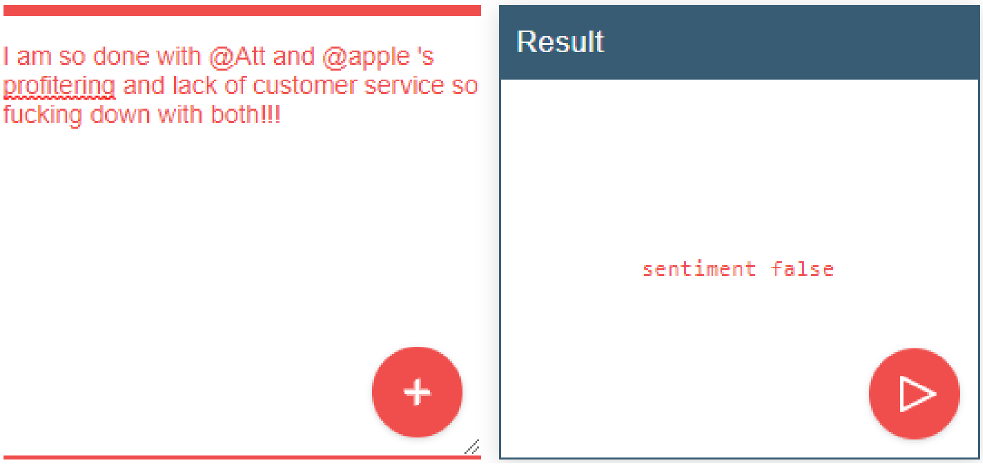


Fig. 6. Application

4.2 Genetic Algorithm For Optimization

A genetic algorithm is a heuristic approach that is inspired by theory of natural selection. This algorithm follow the process of natural selection where the fittest (best) individuals are selected for reproduction in order to produce offspring of the next generation.By saying, genetic algorithms are an optimization algorithm, it means, they are used to find the optimal solution(s) to a given computational problem that maximizes or minimizes a particular function.

Since genetic algorithms are designed to simulate a biological process called evolution, much of the relevant terminology is borrowed from biology. However, the entities that this terminology refers to in genetic algorithms are much simpler than their biological counterparts. The basic components common to almost all genetic algorithms are:

1. Population of chromosomes
2. Crossover to produce next generation of chromosomes
3. Mutation for change in next generation
4. Fitness function for optimal solution
5. Selection

In our project, we have implemented a hybrid model of Neural network and optimize the network/features using Genetic algorithms. As genetic algorithm begins with a randomly chosen assortment of chromosomes, which serves as the first generation (initial population). In this approach **Population** is multiple networks with different weights.

$$chromosomes = [N_1, N_2, N_3, \ldots, N_n]$$

Then each chromosome in the population is evaluated by the fitness function to test how well it solves the problem at hand. So, **fitness** function is the model's accuracy. Now the selection operator chooses some of the chromosomes for reproduction based on a probability distribution defined by the user. The fitter a network is, the more likely it is to be selected. Mutation is sort of applied randomly to check the accuracy measure on any mutated model. So, model's weights are tweaked with the mutation ratio which is 0.05 in this case. Figure 7 is showing the flow of the networks along with the evolution.

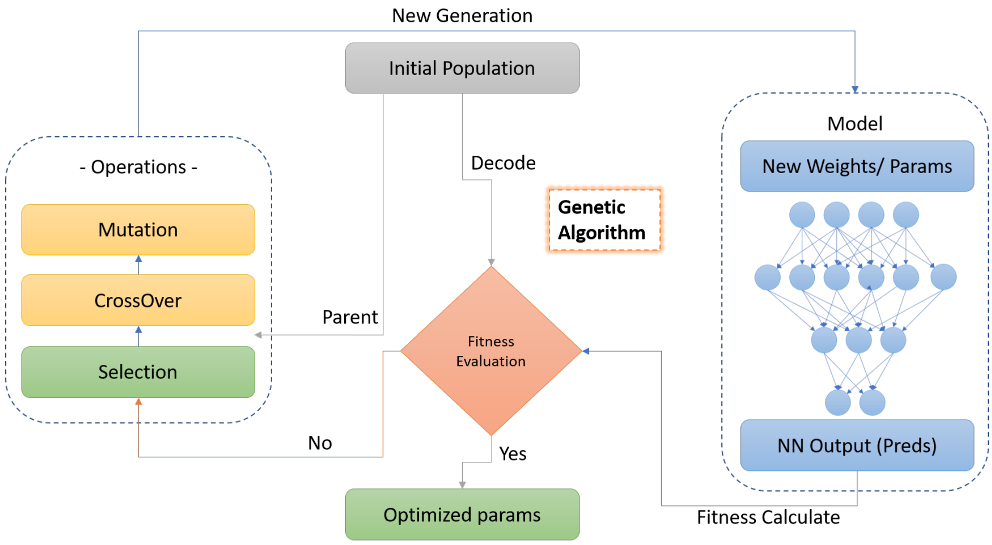


Fig. 7. Genetic Algorithm

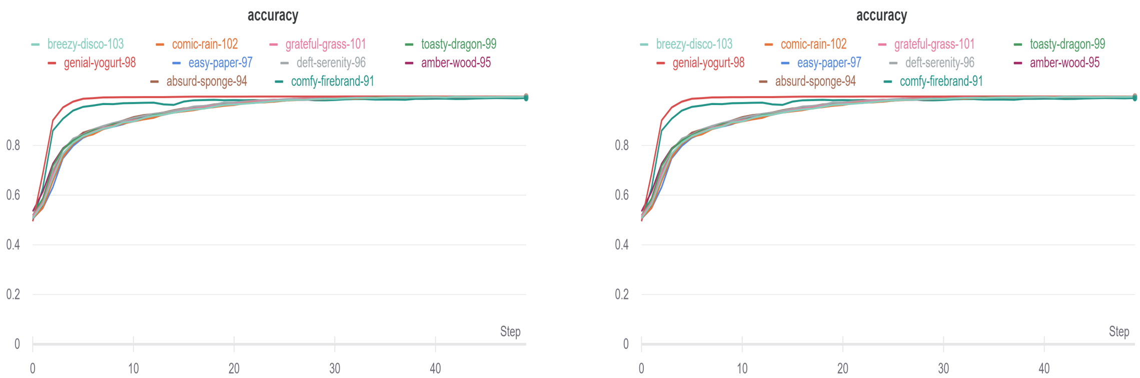


Fig. 8. GA Accuracy on IMDB Dataset

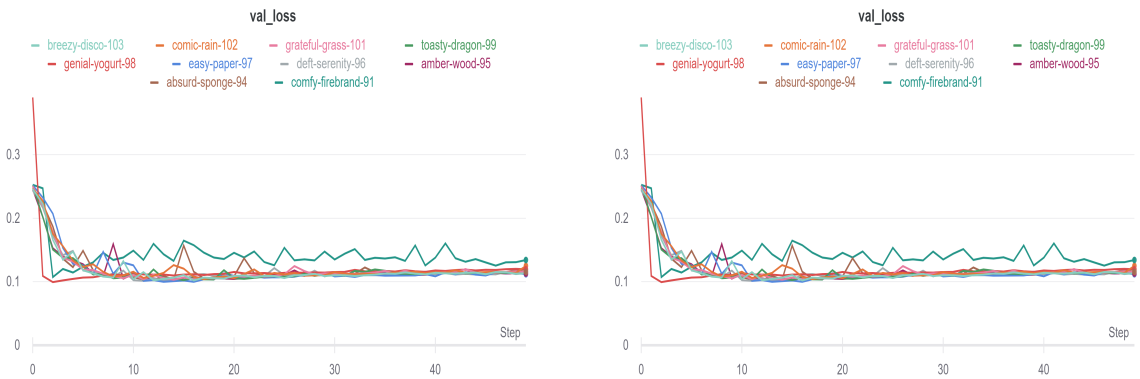


Fig. 9. GA Loss on IMDB Dataset

So, for Optimization there were 80-100 models trained using Genetic Algorithm which took approximately 6-8 hours. We have observed that using GA for random weights of the model can increase the accuracy. So due to time constrains, we used the cnn network only with some customized layer as LSTM will take more time then that.

5 Evaluation

In this section, the results are shown on all models.

5.1 Dataset

The dataset, we are using for our research project consists of the IMDB reviews. It is a dataset of comments from audiences about the stories in films. It contains 50,000 such reviews. Some of the data insights are as follows figure 10

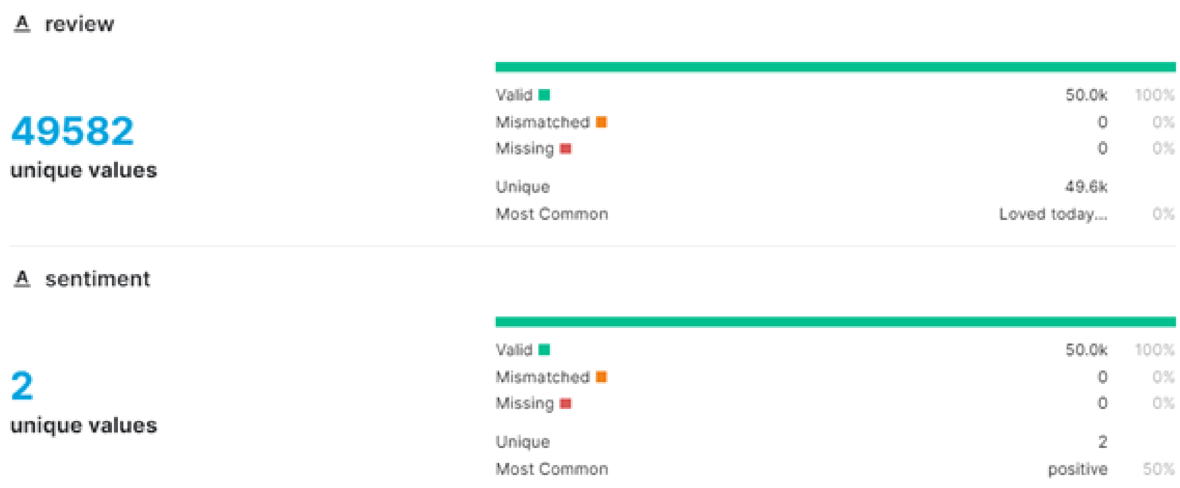


Fig. 10. Dataset Stat

5.2 Results

As models are trained on IMDB dataset. After applying the data cleaning 3.1, we split the dataset into 70:30 ratio and train the models accordingly. The results we have achieved so far on **IMDB** are given in table 1 and **Twitter** are given in table 2

Table 1
Our Project's Findings On IMDB

Models	Parameters	Train Accuracy	Test Accuracy
LSTM	0.558M	97.90%	88.10%
Bidirectional LSTM	3.7M	99.87%	86.74%
CNN	3.2M	98.49%	83.64%
Bert Classification-Model	109M	96.61%	88.24%
TF-Bert For Sequence-Classification	112M	96.47%	90.01%
Hybrid Model	2.5M	94.60%	87.72%

Twitter

Comparison with other research work As there are already multiple papers have been presented with highest reported test-accuracy 97% on IMDB data-set. These reported results are the state-of-the-art and it's no comparison with ours in terms of **how computationally**

Table 2
Reported Accuracy of ML Classification Models i.e., Discriminative Models on Twitter dataset

Models	Train Accuracy	Test Accuracy
SVM	89.16%	77.90%
Logistic Regression	82.34%	78.72%
Random Forest	70.30%	69.10%
Naive Bayes	83.86%	77.44%

expensive they are and there are many different perspective solutions there to address the problem. Few of them is following:

Models	Parameters	Train Accuracy	Test Accuracy
LSTM	100 block (EMB)	97.90%	88.10%
BERT Sequence Classifier	112M (No EMB)	96.47%	90.01%
RNN [17]	128 block (EMB)	-	68.64%
CNN [17]	2.6M	-	88.22%
S-LSTM [18]	8,858K	-	87.15%
CNN [19]	Multiword (EMB)	-	88.1%
CNN+LSTM [19]	Multiword (EMB)	-	87.3%
CNN+LSTM [19] Pre-Processed	Multiword (EMB)	-	88.9%

Table 3
Performance Table on different state-of-the-art approaches

In Table 3, $-$ is representing, the accuracy is not reported in the papers and EMB is the Embedding.

Analysis

6 Conclusion

Sentiment analysis is the common problem nowadays due to exponential increase in content on the internet. For that purpose we have used the neural networks and fuzzy rule based system for text classification between positive and negative reviews. We have identified the dataset, completed the literature review activity and now working on the methodology along with the implementation. We have run the initial pass on IMDB dataset with three neural network models and preliminary results are showing in the table 1. As for the recommendation application for IMDB dataset where one can get reviews about the specific movie, the model made it easy for users to get recommendations predicted over past reviews. Now we are starting the next phase of the project which is the implementation and evaluation.

7 References

1. Liu, B. *Sentiment Analysis and Opinion Mining* in *Synthesis Lectures on Human Language Technologies* (2012).
2. Pang, B. & Lee, L. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* **2,** 1–135 (Jan. 2008).
3. Li, A. C. Sentiment Analysis for IMDb Movie Review (2019).
4. Habimana Olivier, Y. L. & Li, R. *Sentiment analysis using deep learning approaches: an overview* in *Science China Information Sciences* **63** (2019).
5. Pietramala, A., Policicchio, V. L., Rullo, P. & Sidhu, I. *A Genetic Algorithm for Text Classification Rule Induction* in *Machine Learning and Knowledge Discovery in Databases* (eds Daelemans, W., Goethals, B. & Morik, K.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008), 188–203.
6. Bidi, N. & Elberrichi, Z. *Feature selection for text classification using genetic algorithms* in *2016 8th International Conference on Modelling, Identification and Control (ICMIC)* (2016), 806–810.
7. Medhat, W., Hassan, A. & Korashy, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* **5,** 1093–1113 (2014).
8. Iqbal, F., Hashmi, J. M., Fung, B. C. M., Batool, R., Khattak, A. M., Aleem, S. & Hung, P. C. K. A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction. *IEEE Access* **7,** 14637–14652 (2019).
9. Dang, N. C., Moreno-García, M. N. & De la Prieta, F. Sentiment Analysis Based on Deep Learning: A Comparative Study. *Electronics* **9** (2020).
10. Zhang, Y., Liu, Q. & Song, L. Sentence-State LSTM for Text Representation. *CoRR* **abs/1805.02474.** arXiv: `1805.02474` (2018).
11. Tang, D., Wei, F., Qin, B., Yang, N., Liu, T. & Zhou, M. Sentiment Embeddings with Applications to Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* **28,** 496–509 (2016).
12. Yin, Y., Song, Y. & Zhang, M. *Document-Level Multi-Aspect Sentiment Classification as Machine Comprehension* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, Sept. 2017), 2044–2054.

13. Basiri, M. E., Nemati, S., Abdar, M., Cambria, E. & Acharya, U. R. ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. *Future Generation Computer Systems* **115,** 279–294 (2021).

14. Pandey*, A. C. & Rajpoot, D. S. Improving Sentiment Analysis using Hybrid Deep Learning Model. *Recent Advances in Computer Science and Communications* **13,** 627–640 (2020).

15. Salur, M. U. & Aydin, I. A Novel Hybrid Deep Learning Model for Sentiment Classification. *IEEE Access* **8,** 58080–58093 (2020).

16. Greenholt, V. How to fine-tune BERT on text classification task.

17. Peng Cen, K. Z. & Zheng, D. Sentiment Analysis Using Deep Learning Approach. *Journal on Artificial Intelligence* **2,** 17–27 (2020).

18. Zhang, Y., Liu, Q. & Song, L. *Sentence-State LSTM for Text Representation* in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, Melbourne, Australia, July 2018), 317–327.

19. Camacho-Collados, J. & Pilehvar, M. T. *On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis* in *BlackboxNLP@EMNLP* (2018).

A Google Drive Link

There are five implementations availble at drive you can find then at here:
https://drive.google.com/drive/folders/1wfge88Z6QLrzwKCFRMCUX-6cM0rH-5Mn?usp=sharing